DC-SCM LTPI IP

IP Version: v2.1.0

User Guide

FPGA-IPUG-02200-2.1

December 2025

Disclaimers

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

Inclusive Language

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

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ACK	Acknowledgement
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ATM	Asynchronous Transfer Mode
BMC	Baseboard Management Controller
CCITT	International Telegraph and Telephone Consultative Committee (French: Comité Consultatif International Téléphonique et Télégraphique)
CPLD	Complex Programmable Logic Device
CRC	Cyclic Redundancy Check
CSR	Control and Status Registers
DC-SCI	Datacenter-ready Secure Control Interface
DC-SCM	Datacenter-ready Secure Control Module
DDR	Double Data Rate
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
GDDR	Generic Double Data Rate
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HDL	Hardware Description Language
HEC	Header Error Control
HPM	Host Processor Module
HPM FPGA	Host Processor Module FPGA
I/O	Input/Output
I2C	Inter-Integrated Circuit
ID	Identification
IF	Interface
IP	Intellectual Property
LL GPIO	Low Latency General Purpose Input/Output
LSCC	Lattice Semiconductor Corporation
LTPI	Low Voltage Differential Signaling Tunneling Protocol and Interface
LVDS	Low Voltage Differential Signaling
MCSI	Multi-Channel Serial Interface
MCTP	Management Component Transport Protocol
MFR	Manufacturer
MSB	Most Significant Bit
N/A	Not Applicable
NAK	No Acknowledgement
NL	Normal Latency
NL GPIO	Normal Latency General Purpose Input/Output
OCP	Open Compute Project
OEM	Original Equipment Manufacturer
OPN	Ordering Part Number
PHY	Physical
Rx	Receiver

Abbreviation	Definition
SCM	Secure Control Module
SDR	Single Data Rate
SGPIO	Serial General Purpose Input/Output
SMBus	System Management Bus
TDM	Time Division Multiplexing
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter

1. Introduction

The Lattice Semiconductor DC-SCM LVDS Tunneling Protocol and Interface (LTPI) IP Core is an Open Compute Project® (OCP) Data Center – Secure Control Module (DC-SCM) Standards compatible solution which is introduced in the DC-SCM 2.0 Specification. This DC-SCM LTPI IP is OCP Ready™.

LTPI is a protocol and interface designed for tunneling various low-speed signals between Host Processor Module (HPM) and Secure Control Module (SCM). The LTPI protocol goes over the LVDS (Low Voltage Differential Signals) electrical interfaces supported by majority of the CPLDs and FPGAs. This is the next generation protocol for DC-SCM 2.0 that serves as a replacement for two Serial GPIO (SGPIO) interfaces.

The LVDS interface provides higher bandwidth and better scalability than the SGPIO interface. It allows for tunneling of not only GPIOs but also low-speed serial interfaces such as I2C and UART. It is also extensible with additional proprietary OEM interfaces and provides support for raw Data tunneling between HPM and SCM CPLDs. It also provides a solution for minimal wire connection between two FPGAs.

1.1. Quick Facts

Table 1.1 presents a summary of the DC-SCM LTPI IP.

Table 1.1. DC-SCM LTPI IP Quick Facts

IP Requirements	Supported Devices	MachXO3™, MachXO3D™, Mach™-NX, MachXO5™-NX, Certus™-NX, CertusPro™-NX, CrossLink™-NX, and MachXO4™.	
	IP Changes[1]	For a list of changes to the IP, refer to the DC-SCM LTPI IP Release Notes (FPGA-RN-02021).	
Resource Utilization	Supported User Interface	GPIO, I2C, UART, OEM, Data to LVDS and vice versa	
	Resources	See Appendix A. Resource Utilization.	
Design Tool Support	Lattice Implementation	MachXO3, MachXO3D, and Mach-NX	IP Core v2.1.0 – Lattice Diamond™ Software 3.14 and Lattice Propel™ Builder 2025.2
		MachXO5-NX, Certus-NX, CertusPro-NX, CrossLink-NX, and MachXO4	IP Core v2.1.0 – Lattice Radiant™ Software 2025.2 and Lattice Propel Builder 2025.2
	Synthesis	Lattice Synthesis Engine	
		Synopsys Synplify Pro® for Lattice	
	Simulation	For a list of supported simulators, see the Lattice Radiant and Lattice Diamond software user guides.	

Note:

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

1.2. Features

The key features of the DC-SCM LTPI IP include:

- Compliant with the DC-SCM 2.1 LTPI revision 1.1, version 1.1 specification
- Link initialization, discovery, and negotiation
- Supports Multi-Channel Serial Interface
- Supports LVDS and subLVDS
- Supports up to five channels aggregation/disaggregation in total
- Supports GPIO, I2C, UART, OEM, and Data channel aggregation
- For I2C interface, each can be configured as Controller or Target
- Supports up to 800 Mbps LVDS data rate for MachXO3™ and Mach-NX family devices
- Supports up to 1200 Mbps LVDS data rate for MachXO5-NX family devices
- Supports AMBA 3 APB Protocol v1.0 for register access of the soft IP and Data Channel
  - Supports PREADY, a ready signal to indicate completion of an APB transfer
  - PSLVERR is only supported in Data Channel related access (error signal indicating failure of a transfer)

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal names that end with:

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

1.3.3. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Descriptions

2.1. Overview

High-level block diagram of the DC-SCM LTPI IP is shown in Figure 2.1. DC-SCM LTPI uses Time Division Multiplexing (TDM) of high speed LVDS full-duplex link to transmit and receive LTPI channels between SCM and HPM. Data received from external channels, usually mapped to a specific physical interface such as GPIO, I2C, and UART, are aggregated and transmitted from one side, for example, SCM, and de-aggregated and remapped back to physical interface in the other side for example, HPM, and vice versa through Low Voltage Differential Signaling (LVDS).

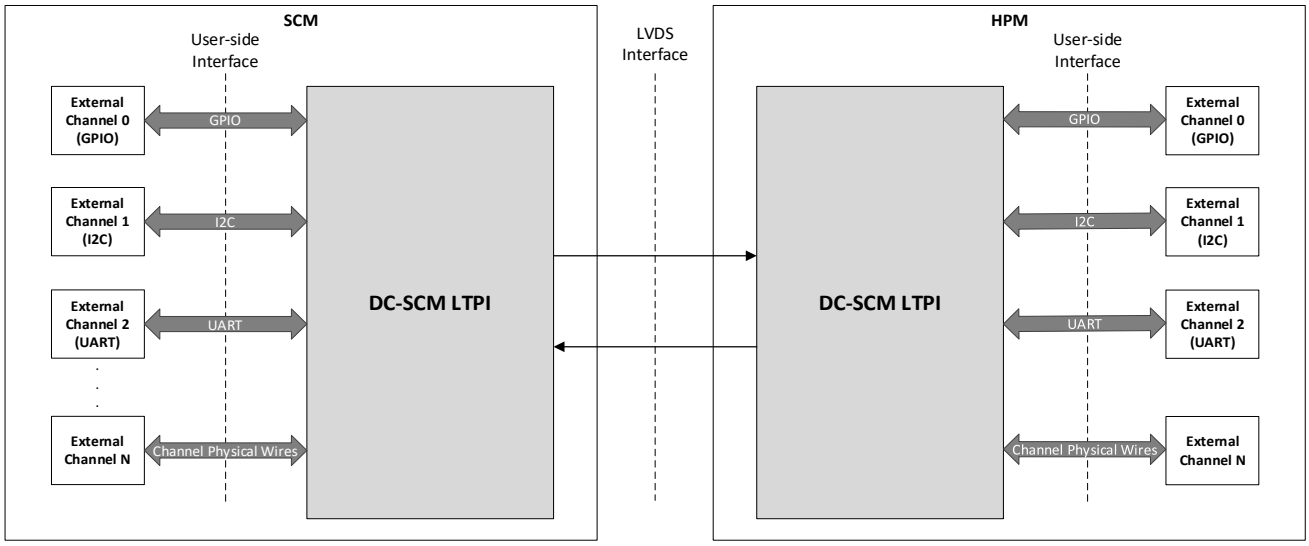


Figure 2.1. High-level Block Diagram

The functional block diagram of the DC-SCM LTPI IP set as either SCM or HPM is shown in Figure 2.2.

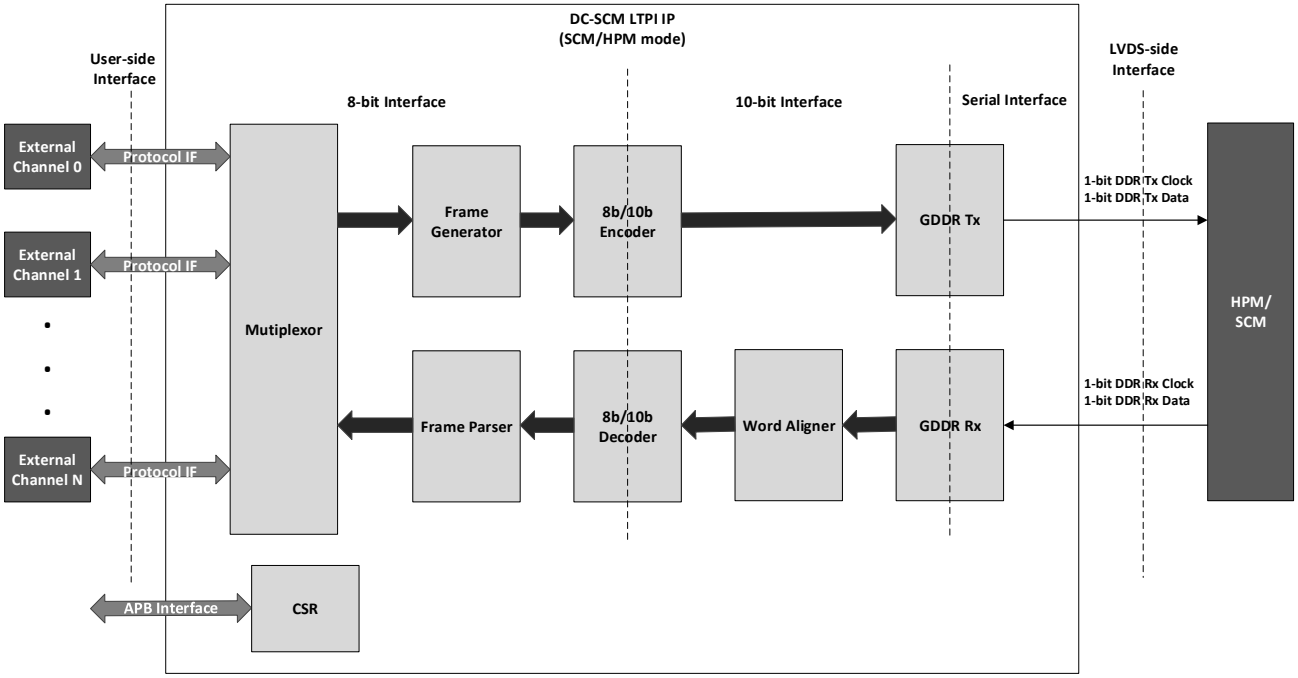


Figure 2.2. Functional Block Diagram of DC-SCM LTPI Set as Either SCM or HPM

Figure 2.3 shows a complete end-to-end block diagram of the SCM to HPM modules.

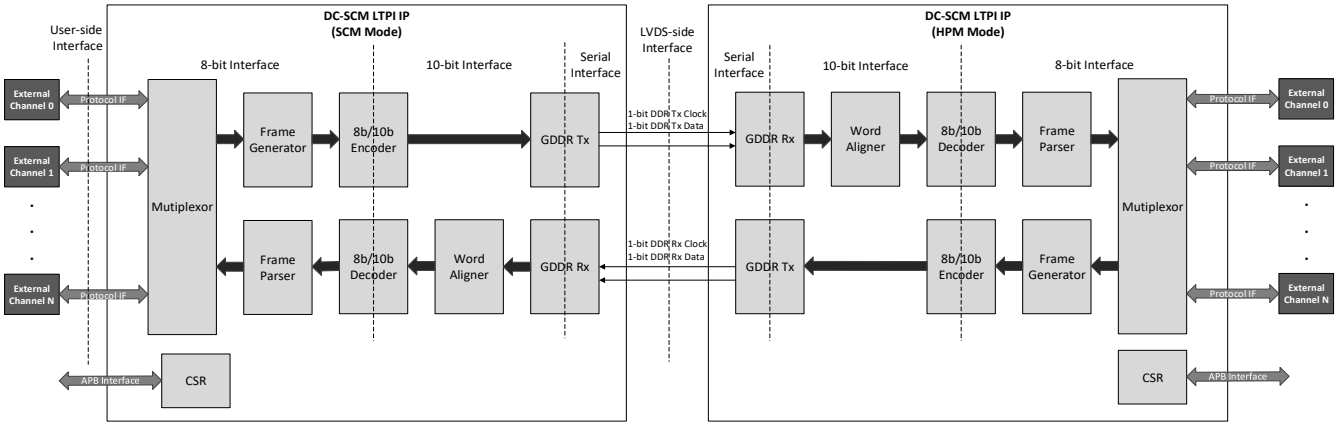


Figure 2.3. DC-SCM LTPI End-to-End SCM to HPM Connection Functional Block Diagram

The DC-SCM LTPI IP consists of Multiplexor, Frame Generator/Parser, 8b/10b Encoder/Decoder, Word Aligner/Link Synchronizer, and GDDR Transmit and Receive modules. In transmit mode, the direction of data flow is from Multiplexor to GDDR Tx. In receive mode, the data path is from GDDR Rx to Multiplexor. An instance of the IP has both Rx and Tx paths available that work simultaneously.

2.2. Signal Description

Table 2.1. DC-SCM LTPI IP Core Signal Description

Port Name	I/O	Width	Default Value	Description
System				
eclk_i	I	1	N/A	GDDR fast sampling clock. Must be the same frequency as the LVDS Clock (MHz).
eclk90_i	I	1	N/A	90-degree phase shifted with respect to eclk_i and is used for transmitting clock generation. Must be source synchronous with eclk_i and of the same frequency as the LVDS Clock (MHz).
clk_i	I	1	N/A	Input system clock used to clock most of the IP core logic. Minimum Frequency is equivalent to: (*LVDS Clock* (MHz) / PHY_MODE) Where: • PHY_MODE = 10 (if SDR only) • PHY_MODE = 5 (if DDR is enabled) To simplify the clocking, the system clock does not need to be exact as per the equation above. The LVDS clock may have some tolerance, so it is better to have a system clock frequency that is higher than the minimum required to avoid frame drops due to FIFO overflow. Below are examples of recommended system clock frequency with the corresponding LVDS Clock range: • SDR Only: **Max LVDS Clock (MHz) / System Clock (MHz)** 200 (and below) / 25 400 (and below) / 50 600 (and below) / 100 • DDR Enabled: **Max LVDS Clock (MHz) / System Clock (MHz)** 100 (and below) / 25 200 (and below) / 50 400 (and below) / 100 600 (and below) / 125
rst_n_i	I	1	N/A	Asynchronous active-low system reset.
int_o	O	1	1'b0	Interrupt signal.
lnk_init_done_o	O	1	1'b0	Indicates that LTPI link training and negotiation are done, and the IP is in active state.
lnk_err_o	O	1	1'b0	Indicates that the IP has encountered a link error. When asserted, it generally means that link training must be performed again. Refer to Figure 2.8 for more details.
Clock Control[1]				
phy_clk_valid_i	I	1	N/A	Indicates that the input clocks eclk_i and eclk90_i are valid and stable. When a clock frequency update is requested (phy_clk_upd_en_o=1), phy_clk_valid_i must de-assert for at least one clock cycle and then assert once the clock is stable (such as when the PLL locks).

Port Name	I/O	Width	Default Value	Description
phy_clk_upd_en_o	O	1	1'd0	Indicates a request to update the clock frequency of eclk_i and eclk90_i to the rate specified by phy_clk_rate_o. The phy_clk_valid_i must de-assert to indicate acknowledgement of the request.
phy_clk_rate_o	O	4	4'd0	Indicates the requested clock frequency rate of eclk_i and eclk90_i: • 4'd0 – 25 MHz (default) • 4'd1 – 50 MHz • 4'd2 – 75 MHz • 4'd3 – 100 MHz • 4'd4 – 150 MHz • 4'd5 – 200 MHz • 4'd6 – 250 MHz • 4'd7 – 300 MHz • 4'd8 – 400 MHz • 4'd9 – 600 MHz • Other values – reserved
LVDS Interface				
lvds_tx_clk_o	O	1	N/A	Differential Tx PHY clock.
lvds_tx_data_o	O	1	N/A	Differential Tx PHY data.
lvds_rx_clk_i	I	1	N/A	Differential Rx PHY clock.
lvds_rx_data_i	I	1	N/A	Differential Rx PHY data.
Low Latency GPIO Channel Interface[2]				
ll_gpio_i	I	*LL GPIO Input Data Width*	N/A	GPIO input pins.
ll_gpio_o	O	*LL GPIO Output Data Width*	{*LL GPIO Output Data Width*{1'b1}}	GPIO output pins.
Normal Latency GPIO Channel Interface[3]				
nl_gpio_i	I	*NL GPIO Input Data Width*	N/A	GPIO input pins.
nl_gpio_o	O	*NL GPIO Output Data Width*	{*NL GPIO Output Data Width* {1'b1}}	GPIO output pins.
I2C Channel Interface[4]				
i2c_scl_io	I/O	*Total Number of I2C/SMBus interface*	N/A	I2C SCL (clock) pin. This is available only if *Enable IO Primitive* is selected. **Note:** [Target] is connected to External I2C Controller, [Controller] is connected to External I2C Target. Bus arrangement of the LTPI I2C/SMBus interface: • SCM: {[Controller], [Target], [Bi-directional]} • HPM: {[Target], [Controller], [Bi-directional]} When bidirectional interface > 0, the bidirectional I2C interfaces are on the lower index of this bus. For example: • Number of LTPI bidirectional interface = 2 • Number of LTPI target interface = 2 • Number of LTPI controller interface = 1 SCM mode: • i2c_scl_io[1:0] are signals for bidirectional I2C channels 0 and 1. • i2c_scl_io[3:2] are signals for I2C target (external I2C controller) channels 0 and 1.

Port Name	I/O	Width	Default Value	Description
				• i2c_scl_io[4] is the signal for I2C controller (external I2C target) channel 0. HPM mode: • i2c_scl_io[1:0] are signals for bidirectional I2C channels 0 and 1. • i2c_scl_io[3:2] are signals for I2C controller (external I2C target) channels 0 and 1. • i2c_scl_io[4] is the signal for I2C target (external I2C controller) channel 0. This arrangement order is fixed and the I2C interface shifts to a lower bit position if a particular interface is not present. For example, if there is no bidirectional interface: SCM mode: • i2c_scl_io[1:0] are signals for I2C target (external I2C controller) channels 0 and 1. • i2c_scl_io[2] are signals for I2C controller (external I2C target) channel 0. HPM mode: • i2c_scl_io[1:0] are signals for I2C controller (external I2C target) channels 0 and 1. • i2c_scl_io[2] are signals for I2C target (external I2C controller) channel 0.
i2c_scl_i	I	*Total Number of I2C/SMBus interface*	N/A	I2C SCL (data) input pin. This is available only if *Enable IO Primitive* is not selected.
i2c_scl_o	O	*Total Number of I2C/SMBus interface*	N/A	I2C SCL (data) output pin. This is available only if *Enable IO Primitive* is not selected.
i2c_scl_oe	O	*Total Number of I2C/SMBus interface*	0	I2C SCL (data) output enable or tri-state control pin. This is available only if *Enable IO Primitive* is not selected.
i2c_sda_io	I/O	*Total Number of I2C/SMBus interface*	N/A	I2C SDA (data) pin. Uses the same mapping as i2c_scl_io. This is available only if *Enable IO Primitive* is selected.
i2c_sda_i	I	*Total Number of I2C/SMBus interface*	N/A	I2C SDA (data) input pin. This is available only if *Enable IO Primitive* is not selected.
i2c_sda_o	O	*Total Number of I2C/SMBus interface*	N/A	I2C SDA (data) output pin. This is available only if *Enable IO Primitive* is not selected.
i2c_sda_oe	O	*Total Number of I2C/SMBus interface*	0	I2C SDA (data) output enable or tri-state control pin. This is available only if *Enable IO Primitive* is not selected.
UART Channel Interface[5]				
uart_tx_i	I	*Number of UART bus interface*	N/A	UART channel input pin. Must be connected to the Tx/Rx bus of UART channel depending on direction.
uart_ctrl_i	I	*Number of UART bus interface*	N/A	UART miscellaneous input pin. Can be connected to CTS/RTS pin depending on direction. Available only if *Enable Flow Control* == Enabled.
uart_rx_o	O	*Number of UART bus interface*	{*Number of UART bus interface*{1'b1}}	UART channel output pin.

Port Name	I/O	Width	Default Value	Description
uart_ctrl_o	O	*Number of UART bus interface*	{*Number of UART bus interface* {1'b1}}	UART miscellaneous output pin. Available only if *Enable Flow Control* == Enabled.
OEM Channel Interface[6]				
oem_io	I/O	*OEM Data Width*	N/A	OEM pins. This is available only if *Enable IO Primitive* is selected.
oem_oe_i	I	*OEM Data Width*	N/A	Output enable of each OEM bidirectional pins. When high, I/O is set as output. When low, I/O is set as input.
oem_i	I	*OEM Data Width*	N/A	OEM input pins. This is available only if *Enable IO Primitive* is not selected.
oem_o	O	*OEM Data Width*	{ *OEM Data Width{1'b1}}*	OEM output pins. This is available only if *Enable IO Primitive* is not selected.
APB Completer Interface (IP CSR + Data Channel[7])				
apb_psel_i	I	1	N/A	Select signal. This indicates that the target device is selected and a data transfer is required.
apb_paddr_i	I	32	N/A	Address signal.
apb_pwdata_i	I	32	N/A	Write data signal.
apb_pwrite_i	I	1	N/A	Direction signal. Write = 1, Read = 0.
apb_penable_i	I	1	N/A	Enable signal. This indicates the second and subsequent cycles of an APB transfer.
apb_pready_o	O	1	1'b0	Ready signal. This indicates transfer completion. Completer uses this signal to extend an APB transfer.
apb_prdata_o	O	32	32'h0	Read data signal.
apb_pslverr_o	O	1	1'b0	APB error signal. Error signal indicating failure of a transfer.
APB Requester Interface (Data Channel[8])				
apb_r_psel_o	O	1	N/A	Select signal. This indicates that the target device is selected and a data transfer is required.
apb_r_paddr_o	O	32	N/A	Address signal.
apb_r_pwdata_o	O	32	N/A	Write data signal.
apb_r_pwrite_o	O	1	N/A	Direction signal. Write = 1, Read = 0.
apb_r_penable_o	O	1	N/A	Enable signal. This indicates the second and subsequent cycles of an APB transfer.
apb_r_pready_i	I	1	1'b0	Ready signal. This indicates transfer completion. Completer uses this signal to extend an APB transfer.
apb_r_prdata_i	I	32	32'h0	Read data signal.
apb_r_pslverr_i	I	1	1'b0	APB error signal. Error signal indicating failure of a transfer.

Notes:

2. Available only if *Enable Programmable Clock Control*==Disabled.
3. Available only if *Enable Low Latency GPIO Channel*==Enabled.
4. Available only if *Enable Normal Latency GPIO Channel*==Enabled.
5. Available only if *Enable I2C Channel*==Enabled.
6. Available only if *Enable UART Channel*==Enabled.
7. Available only if *Enable OEM Channel*==Enabled.
8. Available only if *Enable Data Channel*==Enabled and *IP Mode*==SCM.
9. Available only if *Enable Data Channel*==Enabled and *IP Mode*==HPM and *Data Channel: Enable External Access*==Enabled.

2.3. Attribute Summary

The configurable attributes of the DC-SCM LTPI IP Core are shown and described in this section. The attributes can be configured through the IP Catalog Module/IP wizard of the Lattice Propel Builder software or Lattice Radiant software.

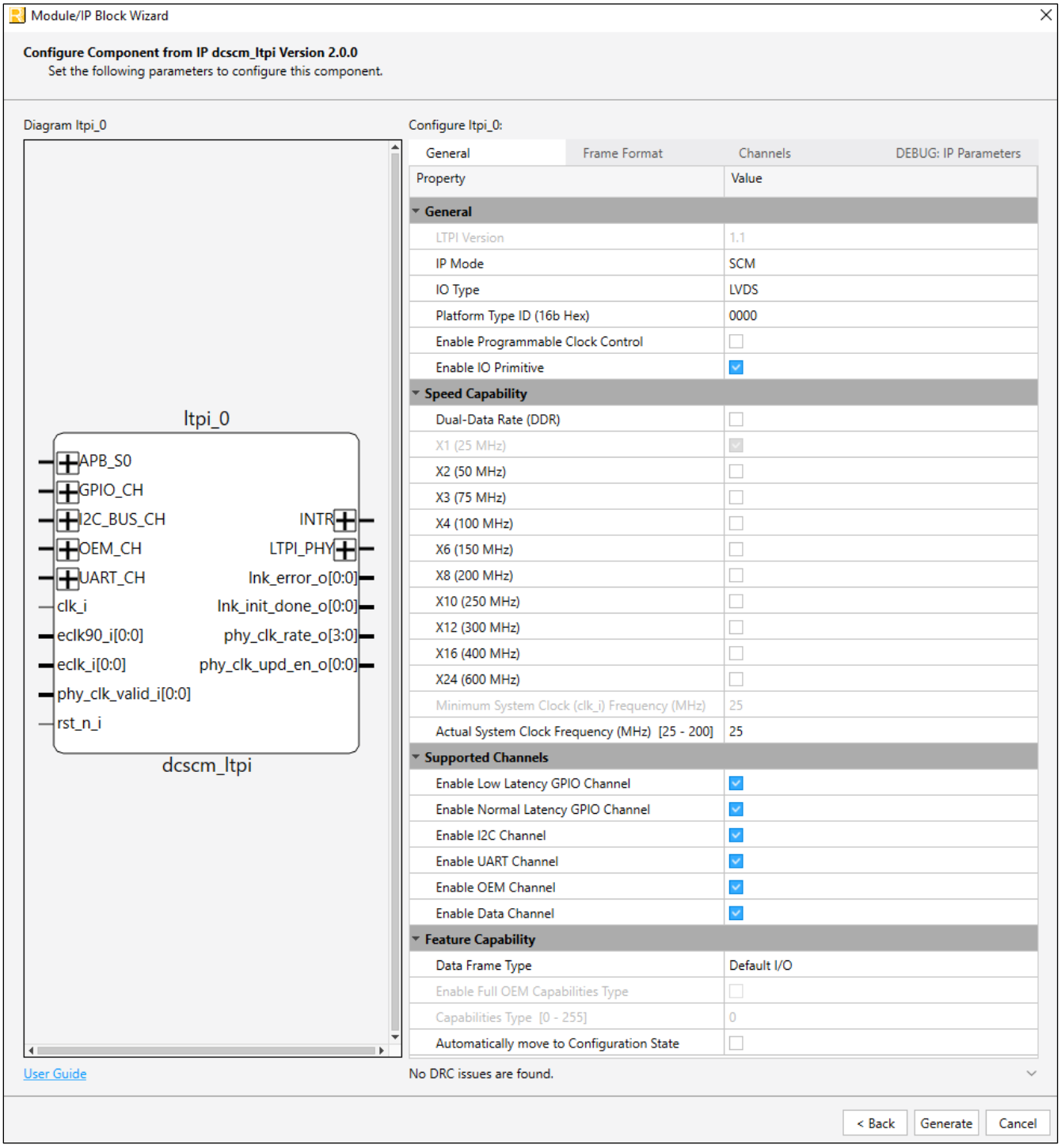


Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core

Table 2.2. Attributes Table for General Tab

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
LTPI Version	—	1.1	Display Only.
IP Mode	SCM, HPM	SCM	—
I/O Type	LVDS, subLVDS[1]	LVDS	—
Platform Type ID (16b Hex)	0000–FFFF	0000	—
Enable Programmable Clock Control	Checked, Unchecked	Unchecked	—
Default: PHY 0 Clock Valid	0, 1	0	Editable only if *Enable Programmable Clock Control* == Checked.
Enable IO Primitive	Checked, Unchecked	Checked	—
Speed Capability			
Dual-Data Rate (DDR)	Checked, Unchecked	Unchecked	—
X1 (25 MHz)	Checked, Unchecked	Checked	Uneditable.
X2 (50 MHz)	Checked, Unchecked	Unchecked	—
X3 (75 MHz)	Checked, Unchecked	Unchecked	—
X4 (100 MHz)	Checked, Unchecked	Unchecked	—
X6 (150 MHz)	Checked, Unchecked	Unchecked	—
X8 (200 MHz)	Checked, Unchecked	Unchecked	—
X10 (250 MHz)	Checked, Unchecked	Unchecked	—
X12 (300 MHz)	Checked, Unchecked	Unchecked	—
X16 (400 MHz)[2]	Checked, Unchecked	Unchecked	—
X24 (600 MHz)	Checked, Unchecked	Unchecked	—
Minimum System Clock (clk_i) Frequency (MHz)	—	25	Display Only.
Actual System Clock Frequency (MHz)	25–200	25	—
Supported Channels			
Enable Low Latency GPIO Channel	Checked, Unchecked	Checked	—
Enable Normal Latency GPIO Channel	Checked, Unchecked	Checked	—
Enable I2C Channel	Checked, Unchecked	Checked	—
Enable UART Channel	Checked, Unchecked	Checked	—
Enable OEM Channel	Checked, Unchecked	Checked	—
Enable Data Channel	Checked, Unchecked	Checked	—
Feature Capability			
Data Frame Type	Custom, Default I/O	Default I/O	—
Enable Full OEM Capabilities Type	Checked, Unchecked	Unchecked	Editable only if *Data Frame Type* == Custom.
Capabilities Type	0–255	0	Viewable and editable only if *Enable Full OEM Capabilities Type* == Checked.
Custom OEM Capability (64b Hex)	—	64'h0	Viewable and editable only if *Enable Full OEM Capabilities Type* == Checked.
Automatically move to Configuration State[3]	Checked, Unchecked	Unchecked	Editable only if *IP Mode* == SCM.

Notes:

1. Available only for MachXO5-NX devices.
2. Maximum speed for MachXO3 and Mach-NX device families.

3. When enabled, it is required for LTPI in SCM and HPM to have a matching feature capabilities during IP configuration. There is no way to manually reconfigure the requested features during the feature negotiation state when this attribute is enabled. Refer to the Request Features section for more details.

Table 2.3. Attributes Description for General Tab

Attribute	Description
General	
LTPI Version	Displays the concatenated major and minor LTPI version.
IP Mode	Indicates if the IP is meant to be used for SCM or HPM. The internal state machine is different between the two modes. For more details, refer to the State Machine section.
I/O Type	Specifies the I/O type for the PHY interface.
Platform Type ID	Two-byte IP Identification Number, which includes GUID, MFR ID, Vendor ID, Board ID, and Revision.
Enable Programmable Clock Control	When enabled, the signals used for requesting clock update are not present and are instead converted to equivalent programmable registers.
Enable IO Primitive	When enabled, the I2C and OEM signals appear as bidirectional ports. Otherwise, the internal tri-state I/O signals are brought out as ports.
Speed Capability	
Dual-Data Rate (DDR)	Indicates support for Dual-Data Rate capability. When enabled, this implies that each X# enabled also supports DDR capability.
X1 (25 MHz)	Indicates support for 25 MHz LVDS Clock.
X2 (50 MHz)	Indicates support for 50 MHz LVDS Clock.
X3 (75 MHz)	Indicates support for 75 MHz LVDS Clock.
X4 (100 MHz)	Indicates support for 100 MHz LVDS Clock.
X6 (150 MHz)	Indicates support for 150 MHz LVDS Clock.
X8 (200 MHz)	Indicates support for 200 MHz LVDS Clock.
X10 (250 MHz)	Indicates support for 250 MHz LVDS Clock.
X12 (300 MHz)	Indicates support for 300 MHz LVDS Clock.
X16 (400 MHz)	Indicates support for 400 MHz LVDS Clock.
X24 (600 MHz)	Indicates support for 600 MHz LVDS Clock.
Minimum System Clock (clk_i) Frequency (MHz)	Shows the recommended minimum system clock frequency.
Actual System Clock Frequency (MHz)	You must indicate the actual clk_i frequency that will be used. This serves as the basis for calculating the necessary values in timer counters.
Supported Channels	
Enable Low Latency GPIO Channel	Enables Low Latency GPIO external channel.
Enable Normal Latency GPIO Channel	Enables Normal Latency GPIO external channel.
Enable I2C Channel	Enables I2C external channel.
Enable UART Channel	Enables UART external channel.
Enable OEM Channel	Enables OEM external channel.
Enable Data Channel	Enables Data external channel. When enabled, APB access that hits the Data Channel address space is translated to data frames.
Feature Capability	
Data Frame Type	Specifies the frame type for data frames. Custom – customizable. Default I/O – predefined frame based on DC-SCM LTPI 1.0 specifications. Refer to the Data/I/O Frames section for more details.
Enable Full OEM Capabilities Type	Enables customized value for *Capabilities Type* that is used in Advertise Frame. This allows the use for non-default and non-IP predefined Capabilities. When enabled, Feature Capability fields of Advertise Frame get value from *Custom OEM Capability (64b Hex).* Refer to the Advertise Frame section for details.

Attribute	Description
Capabilities Type	Indicates if Advertise Frame mapping follows default, IP predefined, or user-customized capabilities mapping. Refer to Table 2.27 for details.
Custom OEM Capability (64b Hex)	Indicates the OEM Feature Capability of the IP. This information is used in Advertise Frame if *Full OEM Capabilities Type* ==Checked. Refer to the Advertise Frame section for details.
Automatically move to Configuration State	Specifies that when *IP Mode* == SCM, the IP automatically goes to Configuration state after completing the required frame transmission for Advertise State. When unchecked, you must perform manual feature request related programming. Refer to the Request Features section for more details.

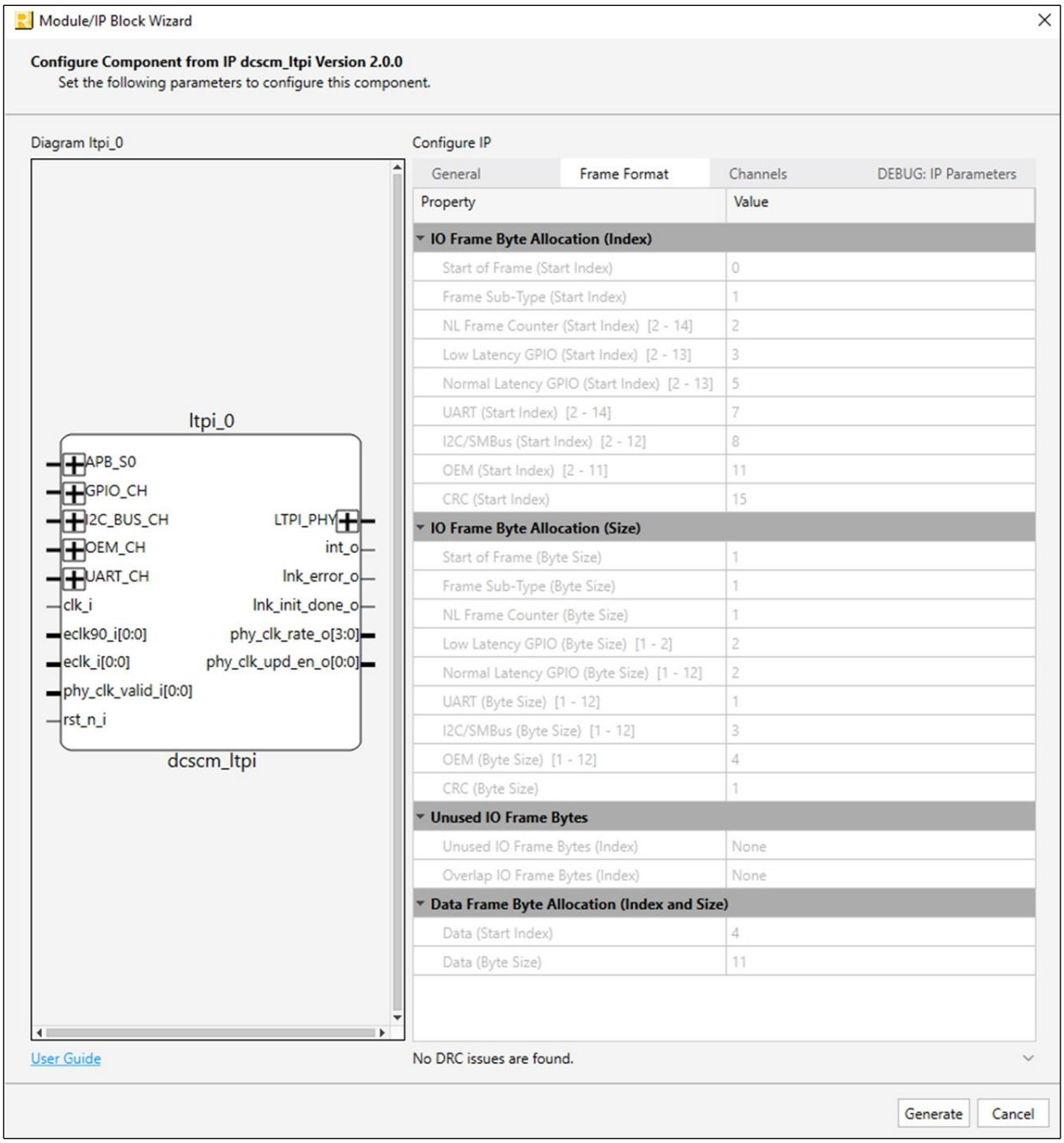


Figure 2.5. Frame Format Tab User Interface of DC-SCM LTPI IP Core

Table 2.4. Attributes Table for Frame Format Tab

Attribute	Selectable Values	Default	Dependency on Other Attributes
I/O Frame Byte Allocation (Index)			
Start of Frame (Start Index)	—	0	—
Frame Sub-Type (Start Index)	—	1	—
NL Frame Counter (Start Index)	2–14	2	Editable only if *Data Frame Type == Custom* and *Enable Normal Latency GPIO Channel == Checked*.
Low Latency GPIO (Start Index)	2–13	3	Editable only if *Data Frame Type == Custom* and *Enable Low Latency GPIO Channel == Checked*.
Normal Latency GPIO (Start Index)	2–13	5	Editable only if *Data Frame Type == Custom* and *Enable Normal Latency GPIO Channel == Checked*.
UART (Start Index)	2–14	7	Editable only if *Data Frame Type == Custom* and *Enable UART Channel == Checked*.
I2C/SMBus (Start Index)	2–12	8	Editable only if *Data Frame Type == Custom*.
OEM (Start Index)	2–11	11	Editable only if *Data Frame Type == Custom* and *Enable UART Channel == Checked*.
CRC (Start Index)	—	15	—
I/O Frame Byte Allocation (Size)			
Start of Frame (Byte Size)	—	1	—
Frame Sub-Type (Byte Size)	—	1	—
NL Frame Counter (Byte Size)	—	1	—
Low Latency GPIO (Byte Size)	1–12	2	Editable only if *Data Frame Type == Custom* and *Enable Low Latency GPIO Channel == Checked*.
Normal Latency GPIO (Byte Size)	1–12	2	Editable only if *Data Frame Type == Custom* and *Enable Normal Latency GPIO Channel == Checked*.
UART (Byte Size)	1–12	1	Editable only if *Data Frame Type == Custom* and *Enable UART Channel == Checked*.
I2C/SMBus (Byte Size)	1–12	3	Editable only if *Data Frame Type == Custom*.
OEM (Byte Size)	1–12	4	Editable only if *Data Frame Type == Custom* and *Enable UART Channel == Checked*.
CRC (Byte Size)	—	1	—
Unused I/O Frame Bytes			
Unused IO Frame Bytes (Index)	—	—	Display only.
Overlap IO Frame Bytes (Index)	—	—	Display only.
Data Frame Byte Allocation (Index and Size)			
Data (Start Index)	—	4	Display only.
Data (Byte Size)	—	11	Display only.

Table 2.5. Attributes Description for Frame Format Tab

Attribute	Description
Frame Format: I/O Frame Byte Allocation	
<Field name> (Start Index)	This is the starting position of the field in the 16-byte Frame with indices ranging from 0 to 15. When *Data Frame Type==Default IO*, this configuration is fixed, and the channels are located at predefined positions as specified in the DC-SCM LTPI Specification. When *Data Frame Type==Custom,* you can customize the location of each channel to suit the desired byte sequence.
<Field name> (Byte Size)	This is the byte size allocation of the field. When *Data Frame Type==Default IO*, this configuration is fixed. When *Data Frame Type==Custom,* you can customize the number of payload bytes allocated to each channel.

Attribute	Description
	Each external channel, when enabled, is mapped to the 13-byte data portion of a Data I/O frame. The byte size allocation determines the maximum number of bus or interface widths of each channel. Refer to the Data/I/O Frames section for details.
Frame Format: Unused I/O Frame Bytes	
Unused IO Frame Bytes (Index)	Displays the byte indices that are free or not allocated.
Overlap IO Frame Bytes (Index)	Displays the byte indices that overlaps with other allocation.

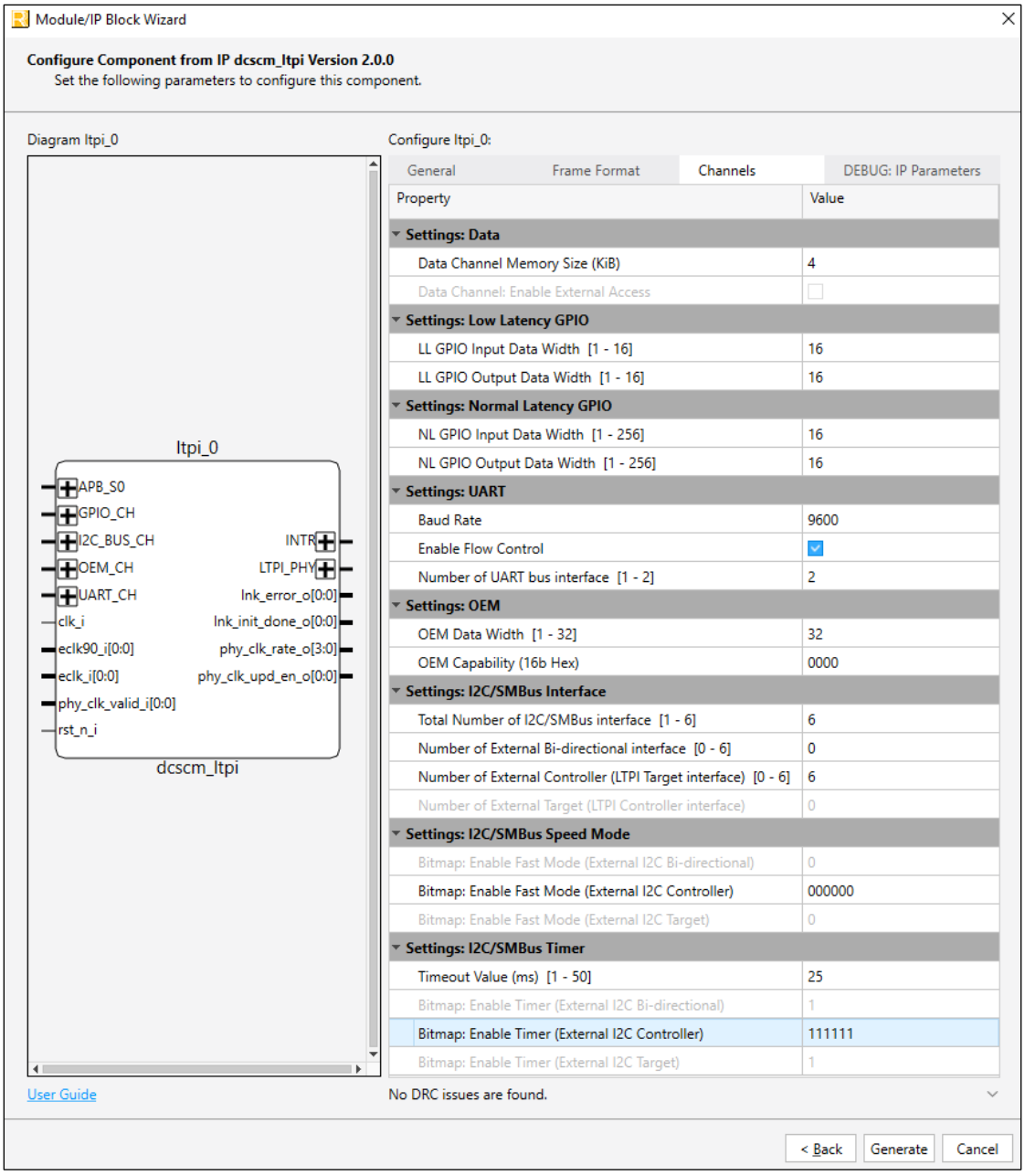


Figure 2.6. Channels Tab User Interface of DC-SCM LTPI IP Core

Table 2.6. Attributes Table for Data

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings			
Data Channel Memory Size (KiB)	4, 8, 16, 32, …, 2097152	4	Editable only if *Enable Data Channel == Checked*.
Data Channel: Enable External Access	Checked, Unchecked	Unchecked	Editable only if *Enable Data Channel == Checked* and *IP Mode == HPM*.

Table 2.7. Attributes Description for Data

Attribute	Description
Settings	
Data Channel Memory Size (KiB)	Specifies the size of the address space accessible to Data Channel. The size is in multiples of 1024 bytes or 1 KiB.
Data Channel: Enable External Access	Enables the APB requestor interface in the HPM for Data Channel external access.

Table 2.8. Attributes Table for LL GPIO

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings			
LL GPIO Input Data Width	1–(8 × *Low Latency GPIO (Byte Size)*)	16	Editable only if *Enable Low Latency GPIO Channel == Checked*.
LL GPIO Output Data Width	1–(8 × *Low Latency GPIO (Byte Size)*)	16	Editable only if *Enable Low Latency GPIO Channel == Checked*.

Table 2.9. Attributes Description for LL GPIO

Attribute	Description
Settings	
LL GPIO Input Data Width	Specifies the number of LL GPIO input pins. The maximum LL GPIO input pins depend on how many bytes are allocated for LL GPIO Channel in the Frame Format configuration (1 byte = 8 pins).
LL GPIO Output Data Width	Specifies the number of LL GPIO output pins. The maximum LL GPIO output pins depend on how many bytes are allocated for LL GPIO Channel in the Frame Format configuration (1 byte = 8 pins).

Table 2.10. Attributes Table for NL GPIO

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings			
NL GPIO Input Data Width	1–256	16	Editable only if *Enable Normal Latency GPIO Channel == Checked*.
NL GPIO Output Data Width	1–256	16	Editable only if *Enable Normal Latency GPIO Channel == Checked*.

Table 2.11. Attributes Description for NL GPIO

Attribute	Description
Settings	
NL GPIO Input Data Width	Specifies the number of NL GPIO input pins. The maximum NL GPIO input pins tunneled in a single frame depend on how many bytes are allocated for NL GPIO Channel in the Frame Format configuration (1 byte = 8 pins). If the width exceeds the allocated payload bytes, then multiple frames are needed to completely transfer the NL GPIO input.

Attribute	Description
NL GPIO Output Data Width	Specifies the number of NL GPIO output pins. The maximum NL GPIO output pins tunneled in a single frame depend on how many bytes are allocated for NL GPIO Channel in the Frame Format configuration. (1 byte = 8 pins). If the width exceeds the allocated payload bytes, then multiple frames are needed to completely transfer the NL GPIO output.

Table 2.12. Attributes Table for I2C

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings: I2C/SMBus Interface			
Total Number of I2C/SMBus interface	0–*(2 × I2C/SMBus (Byte Size)*)	6	Editable only if *Enable I2C Channel == Checked.*
Number External Bi-directional interface	1–*Total Number of I2C/SMBus interface*	0	Editable only if *Enable I2C Channel == Checked.*
Number External Controller (LTPI Target interface)	1–(*Total Number of I2C/SMBus interface – Number External Bi-directional interface*)	6	Editable only if *Enable I2C Channel == Checked.*
Number External Target (LTPI Controller interface)	—	*Total Number of I2C/SMBus interface - Number External Bi-directional interface - Number External Controller (LTPI Target interface)*	Uneditable.
Settings: I2C/SMBus Speed Mode			
Bitmap: Enable Fast Mode (External I2C Bi-directional)	0 or 1 (bitmap per interface)	0	Editable only if *Enable I2C Channel == Checked* and *Number External Bi-directional interface > 0*.
Bitmap: Enable Fast Mode (External I2C Controller)	0 or 1 (bitmap per interface)	000000	Editable only if *Enable I2C Channel == Checked* and *Number External Controller interface > 0*.
Bitmap: Enable Fast Mode (External I2C Target)	0 or 1 (bitmap per interface)	0	Editable only if *Enable I2C Channel == Checked* and *Number External Target interface > 0*.
Settings: I2C/SMBus Timer			
Timeout Value (ms)	1–50	25	Editable only if *Enable I2C Channel == Checked.*
Bitmap: Enable Timer (External I2C Bi-directional)	0 or 1 (bitmap per interface)	1	Editable only if *Enable I2C Channel == Checked* and *Number External Bi-directional interface > 0*.
Bitmap: Enable Timer (External I2C Controller)	0 or 1 (bitmap per interface)	111111	Editable only if *Enable I2C Channel == Checked* and *Number External Controller interface > 0*.
Bitmap: Enable Timer (External I2C Target)	0 or 1 (bitmap per interface)	1	Editable only if *Enable I2C Channel == Checked* and *Number External Target interface > 0*.

Table 2.13. Attributes Description for I2C

Attribute	Description
Settings: I2C/SMBus Interface	
Total Number of I2C/SMBus interface	Specifies the total number of I2C bus interfaces in the I2C channel group. Each I2C bus interface has a dedicated 4-bit data width in the LTPI frame. The maximum number of I2C buses depends on how many bytes are allocated for the I2C channel in the Frame Format configuration (1 byte = 2 I2C bus).

Attribute	Description
	In cases where the number of buses is odd, 0 is padded to complete the payload byte.
Number External Bi-directional interface	Specifies the number of LTPI I2C interface with bidirectional (both target and controller) functions.
Number External Controller (LTPI Target interface)	Specifies the number of LTPI I2C interface with target function only. This interface is connected to an external I2C Controller.
Number External Target (LTPI Controller interface)	Specifies the number of LTPI I2C interface with controller function only. This interface is connected to an external I2C Target.
Settings: I2C/SMBus Speed Mode	
Bitmap: Enable Fast Mode (External I2C Bi-directional/Target/Controller)	Enables fast mode (400 kHz) of the I2C bus. Each bit represents a setting for each I2C interface.
Settings: I2C/SMBus Timer	
Timeout Value (ms)	Specifies the timeout value in milliseconds. Note that the timer counter requires the clock period, which is calculated based on the setting in *Actual System Clock Frequency (MHz)*.
Bitmap: Enable Timer (External I2C Bi-directional/Target/Controller)	Enables the built-in timer in each available I2C bus. Each bus has its own timer logic. When checked, LTPI monitors the SCL bus of I2C channel. Every time SCL bus goes to low, it starts the timeout counter of the corresponding bus. When the timeout threshold is met, the IP automatically performs recovery and resets the internal I2C controller of the bus where timeout is observed, releasing it.

Table 2.14. Attributes Table for UART

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings: UART			
Baud Rate	300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 576000, 921600	9600	Editable only if *Enable UART Channel == Checked*.
Enable Flow Control	Checked, Unchecked	Checked	Editable only if *Enable UART Channel == Checked*.
Number of UART bus interface	1–(2 × *UART (Byte Size)*)	2	Editable only if *Enable UART Channel == Checked*.

Table 2.15. Attributes Description for UART

Attribute	Description
Settings: UART	
Baud Rate	Specifies the UART baud rate.
Enable Flow Control	Enables flow control ports of the UART channel. Automatically set to Disabled if *Enable UART Channel == Disabled*.
Number of UART bus interface	Specifies the number of UART bus interfaces in the UART channel group. The maximum number of UART buses depends on how many bytes are allocated for the UART channel in the Frame Format configuration (1 byte = 2 UART buses). Each UART bus interface has dedicated 4-bit pins for each transmit and receive directions. In cases where the number of bus is odd, *0* is padded to complete the payload byte.

Table 2.16. Attributes Table for OEM

Attribute	Selectable Values	Default	Dependency on Other Attributes
Settings: OEM			
OEM Data Width	1– (8 × *OEM (Byte Size)*)	32	Editable only if *Enable OEM Channel == Checked*.
OEM Capability (16b Hex)	—	0000	Editable only if *Enable OEM Channel == Checked*.

Table 2.17. Attributes Description for OEM

Attribute	Description
Settings: OEM	
OEM Data Width	Specifies the number of OEM pins. The maximum number of OEM pins depends on how many bytes are allocated for the OEM channel in the Frame Format configuration (1 byte = 8 pins).
OEM Capability (16b Hex)	Indicates the default OEM channel capability.

2.4. Register Description

The DC-SCM LTPI soft IP registers are shown in Table 2.20. These are accessed through the APB completer interface of IP CSR.
Note: The unused and/or reserved regions in the address space may return unknown values and this should be ignored.

Table 2.18. Summary of DC-SCM LTPI Register Address Space

Start Offset	End Offset	Size	Description
0x0000	0x00FF	256 bytes	Reserved.
0x0100	0x01FF	256 bytes	LSCC Specific Registers.
0x0200	0x02FF	256 bytes	Control and Status Registers. These are the same registers defined in the DC-SCM LTPI Specification. Some registers, like Advertise Capabilities, may change the field definition depending on the Frame Format. Refer to the Frame Format section for more details.
0x0300	0x0FFF	13 × 256 bytes	Reserved.
0x1000	0x1000+Size	*Data Channel Memory Size*	Data Channel Mapping. This is applicable only if *IP MODE* == *SCM* and *Enable Data Channel* == Checked. APB access to this region is translated to Data Frame and forwarded to the HPM side.

Table 2.19. Access Type Definition

Access Type	Behavior on Read Access	Behavior on Write Access
RO	Returns register value	Ignores write access.
WO	Returns 0	Updates register value.
RW	Returns register value	Updates register value.
RW1C	Returns register value	Writing 1’b1 on register bit clears the bit to 1’b0. Writing 1’b0 on register bit is ignored.
RSVD	Returns 0	Ignores write access.

Table 2.20. DC-SCM LTPI Soft IP Registers

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
Control and Status Registers						
0x00	Link Status	RO	0x0	0	LTPI Link Aligned	1 – Link aligned 0 – Otherwise
		RW1C	0x0	1	LTPI Link Lost Error	1 – Error detected 0 – Otherwise
		RW1C	0x0	2	Frame CRC Error	1 – Error detected 0 – Otherwise
		RW1C	0x0	3	Unknown Comma Symbol Error	1 – Error detected 0 – Otherwise
		RW1C	0x0	4	Link Speed Timeout Error	1 – Error detected 0 – Otherwise
		RW1C	0x0	5	Link Configure/Accept Timeout Error	1 – Error detected 0 – Otherwise
		RO	0x0	6	—	Reserved
		RO	0x0	7	DDR Mode	1 – DDR mode 0 – SDR mode
		RO	0x0	11:8	LTPI Link Speed	0x0 – 25 MHz (default) 0x1 – 50 MHz 0x2 – 75 MHz 0x3 – 100 MHz 0x4 – 150 MHz 0x5 – 200 MHz 0x6 – 250 MHz 0x7 – 300 MHz 0x8 – 400 MHz 0x9 – 600 MHz Other values: Reserved
		RO	0x0	15:12	Remote Link State	0x4 – Operational (L0) Other values – Remote link not in L0
		RO	0xF	19:16	LTPI Link State	0xF – Reset 0x0 – Link detect 0x1 – Link speed 0x2 – Advertise 0x3 – Configuration 0x4 – Operational (L0) Other values: Reserved
		RO	0x0	31:20	—	Reserved
0x04	Detect Capabilities Local	RO	0x1	3:0	LTPI Minor Version	Minor version number
		RO	0x1	7:4	LTPI Major Version	Major version number
		RO	Takes the value from the IP GUI setting *X1 (25 MHz)*.	8	Link Speed Capabilities (25 MHz)	1 – Supported (fixed)
		RW	Takes the value from the IP GUI setting *X2 (50 MHz)*.	9	Link Speed Capabilities (50 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
		RW	Takes the value from the IP GUI setting *X2 (75 MHz)*.	10	Link Speed Capabilities (75 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (100 MHz)*.	11	Link Speed Capabilities (100 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (150 MHz)*.	12	Link Speed Capabilities (150 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (200 MHz)*.	13	Link Speed Capabilities (200 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (250 MHz)*.	14	Link Speed Capabilities (250 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (300 MHz)*.	15	Link Speed Capabilities (300 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X2 (400 MHz)*.	16	Link Speed Capabilities (400 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RW	Takes the value from the IP GUI setting *X24 (600 MHz)*.	17	Link Speed Capabilities (600 MHz)	1 – Supported 0 – Not supported Available only if the corresponding IP GUI setting is selected.
		RO	0x0	22:18	—	Reserved
		RW	Takes the value from the IP GUI setting *Dual-Data Rate (DDR)*.	23	Link Speed Capabilities (DDR)	1 – DDR mode 0 – SDR mode Available only if *Dual-Data Rate (DDR)* is selected.
		RO	0x0	31:24	—	Reserved
0x08	Detect Capabilities Remote	RO	0x0	31:0	Remote link capabilities. Same fields as *Detect Capabilities Local*.	Remote link capabilities. Same fields as *Detect Capabilities Local*.
0x0C	Platform ID Local	RO	Takes the value from the IP GUI setting *Platform Type ID (16b Hex)*.	15:0	SCM/HPM ID	OEM defined ID
		RO	0x0	31:16	—	Reserved
0x10	Platform ID Remote	RO	0x0	31:0	Remote link Platform ID. Same fields as *Platform ID Local*.	Remote link Platform ID. Same fields as *Platform ID Local*.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x14	Advertise Capabilities Local Low[1]	RW	Takes the value from the IP GUI setting *Enable Low Latency GPIO Channel == Checked or Enable Normal Latency GPIO Channel == Checked*.	0	Supported Channel (GPIO)	1 – Supported 0 – Not supported
		RW	Takes the value from the IP GUI setting *Enable I2C Channel == Checked*.	1	Supported Channel (I2C/SMBus)	1 – Supported 0 – Not supported Available only if *Enable I2C Channel == Checked*.
		RW	Takes the value from the IP GUI setting *Enable UART Channel == Checked*.	2	Supported Channel (UART)	1 – Supported 0 – Not supported Available only if *Enable UART Channel == Checked*.
		RW	Takes the value from the IP GUI setting *Enable Data Channel == Checked*.	3	Supported Channel (Data)	1 – Supported 0 – Not supported Available only if *Enable Data Channel == Checked*.
		RW	Takes value from IP GUI setting *Enable OEM Channel == Checked*.	4	Supported Channel (OEM)	1 – Supported 0 – Not supported Available only if *Enable OEM Channel == Checked*.
		RW	0x0	7:5	—	Reserved
		RW	Takes the value from the IP GUI setting *NL GPIO Input Data Width* (or *NL GPIO Output Data Width* if it is higher).	17:8	Number of NL GPIOs	0 – Not supported 1 – 1 GPIO 2 – 2 GPIO … Up to the max setting of *NL GPIO Input Data Width* (or *NL GPIO Output Data Width* if it is higher). Available only if *Enable Low Latency GPIO Channel == Checked or Enable Normal Latency GPIO Channel == Checked*.
		RW	0x0	23:18	—	Reserved
		RW	Takes the value from the IP GUI setting *Total Number of I2C/SMBus interface > 0*.	24	I2C/SMBus Ch 0 Enable	1 – Enabled 0 – Disabled Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 0*.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
		RW	Takes the value from the IP GUI setting *Total Number of I2C/SMBus interface > 1*.	25	I2C/SMBus Ch 1 Enable	1 – Enabled 0 – Disabled Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 1*.
		…	…	…	…	…
		RW	Takes the value from the IP GUI setting *Total Number of I2C/SMBus interface > 5*.	29	I2C/SMBus Ch 5 Enable	1 – Enabled 0 – Disabled Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 5*.
		RO	Same value as *Supported Channel (I2C/SMBus)*.	30	Echo Support	Always set to 1 if *Enable I2C Channel == Checked*.
		RO	0x0	31	—	Reserved
0x18	Advertise Capabilities Local High[1]	RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*.	0	I2C/SMBus 0 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 0*.
		RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*.	1	I2C/SMBus 1 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 1*.
		RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*.	2	I2C/SMBus 2 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 2*.
		RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*.	3	I2C/SMBus 3 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 3*.
		RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*.	4	I2C/SMBus 4 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 4*.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
		RW	Takes the value from the IP GUI setting *Bitmap: Enable Fast Mode*	5	I2C/SMBus 5 Speed	1 – 400 kHz 0 – 100 kHz Available only if *Enable I2C Channel == Checked* and *Total Number of I2C/SMBus interface > 5*.
		RO	0x0	7:6	—	Reserved
		RW	Takes the value from the IP GUI setting *Baud Rate*.	11:8	UART Max Baud Rate	For more details, see Table 2.31. Available only if *Enable UART Channel == Checked*.
		RW	Takes the value from the IP GUI setting *Enable Flow Control*.	12	Flow Control	1 – Enabled 0 – Disabled Available only if *Enable UART Channel == Checked*.
		RW	Takes the value from the IP GUI setting *Number of UART bus interface > 0*.	13	UART 0 Enable	1 – Enabled 0 – Disabled Available only if *Enable UART Channel == Checked*.
		RW	Takes the value from the IP GUI setting *Number of UART bus interface > 1*.	14	UART 1 Enable	1 – Enabled 0 – Disabled Available only if *Enable UART Channel == Checked*.
		RO	0x0	15	—	Reserved
		RW	Takes the value from the IP GUI setting *OEM Capability (16b Hex)*.	31:16	OEM Capabilities	Available only if *Enable OEM Channel == Checked*.
0x1C	Advertise Capabilities Remote Low	RO	0x0	31:0	Remote link capabilities. Same fields as *Advertise Capabilities Local Low*.	Remote link capabilities. Same fields as *Advertise Capabilities Local Low*.
0x20	Advertise Capabilities Remote High	RO	0x0	31:0	Remote link capabilities. Same fields as *Advertise Capabilities Local High*.	Remote link capabilities. Same fields as *Advertise Capabilities Local High*.
0x24	Default Configuration Low	RO	Same as *Advertise Capabilities Local Low*	31:0	Default local configuration. Same fields as *Advertise Capabilities Local Low*.	Default local configuration. Same fields as *Advertise Capabilities Local Low*.
0x28	Default Configuration High	RO	Same as *Advertise Capabilities Local High*	31:0	Default local configuration. Same fields as *Advertise Capabilities Local High*.	Default local configuration. Same fields as *Advertise Capabilities Local High*.
0x2C–0x7F	Reserved	RO	N/A	N/A	N/A	N/A

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x80	Link Control	RW	0x0	0	LTPI Link Software Reset	1 – Reset asserted 0 – Reset de-asserted Resets only the interface channels and returns the Link to Advertise State. Reset does not affect the GPIO and OEM output signals.
		RW	0x0	1	LTPI Link Retraining Request	1 – Reset asserted 0 – Reset de-asserted Resets the IP Core (except the CSR) and restarts the link training. Reset does not affect the GPIO and OEM output signals.
		RW	0x0	6:2	LTPI I2C Channel Reset (channel ID)	0x00 – I2C Ch0 0x01 – I2C Ch1 … 0x18 – I2C Ch23 Other values: Reserved Available only if *Enable I2C Channel == Checked*.
		RW	0x0	7	LTPI I2C Channel Reset (all I2C)	1 – Selected I2C channel only 0 – All I2C channels Available only if *Enable I2C Channel == Checked*.
		RW	0x0	8	LTPI I2C Channel Reset (trigger)	1 – Reset asserted 0 – Reset de-asserted Available only if *Enable I2C Channel == Checked*.
		RW	0x0	9	Data Channel Reset	1 – Reset asserted 0 – Reset de-asserted Available only if *Enable Data Channel == Checked*.
		RO	Takes the value from the IP GUI setting *Automatically move to Configuration State*.	10	Auto Move to Configuration State	1 – Automatically move the SCM to Configuration State 0 – Wait for Trigger to Configuration State (bit[11])
		RW	0x0	11	Trigger to Configuration State	1 – Enables the SCM to move to Configuration State Resets when *Link Software Reset* (Link Control [0]) or *Link Retraining Request* (Link Control [1]) is asserted. Available only if *IP Mode == SCM* and *Automatically move to Configuration State* is not selected.
		RO	0x0	31:12	—	Reserved
0x84–0xFF	Reserved	RO	—	—	—	N/A
LSCC Specific Registers						
0x00	Interrupt Status	RO	0x0	4:0	—	Reserved
		RW1C	0x0	5	Rx FIFO Overflow Detected	1 – Error detected

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
		RW1C	0x0	6	Tx FIFO Underflow Detected	1 – Error detected
		RW1C	0x0	7	I2C Timeout Detected	1 – Error detected Available only if *Enable I2C Channel == Checked*.
		RW1C	0x0	8	I2C Code Error	1 – Error detected Available only if *Enable I2C Channel == Checked*.
		RO	0x0	15:9	—	Reserved
		RW1C	0x0	16	PHY Clock Update	1 – PHY clock can be updated (Refer to the *Clock Control* register) This is cleared once the *Clock Control* register *PHY Clock Valid* is set to 1 again. Available only if *Enable Programmable Clock Control == Checked*.
		RO	0x0	31:17	—	Reserved
0x04	Interrupt Enable	RW	0x0	0	Enable INT for LTPI Link Lost Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	1	Enable INT for Frame CRC Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	2	Enable INT for Unknown Comma Symbol Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	3	Enable INT for Link Speed Timeout Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	4	Enable INT for Link Configure/Accept Timeout Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	5	Enable INT for Rx FIFO Overflow Detected	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	6	Enable INT for Tx FIFO Underflow Detected	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion
		RW	0x0	7	Enable INT for I2C Timeout Detected	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion Available only if *Enable I2C Channel == Checked*.
		RW	0x0	8	Enable INT for I2C Code Error	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion Available only if *Enable I2C Channel == Checked*.
		RO	0x0	15:9	—	Reserved
		RW	0x0	16	Enable INT for PHY Clock Update	1 – Enable interrupt port assertion 0 – Disable interrupt port assertion Available only if *Enable Programmable Clock Control == Checked*.
		RO	0x0	31:17	—	Reserved
0x08	Interrupt Set	WO	—	31:0	—	For debugging. Force the trigger of the corresponding interrupt status bit.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x0C	Clock Control	RW	0x0	0	PHY Clock Valid 0	0 – PHY Clock is not valid. Holds the LTPI Link process. 1 – PHY Clock is valid. Indicates that the input clocks *eclk_i* and *eclk90_i* are valid and stable. LTPI Link process does not proceed if this is low. Available only if *Enable Programmable Clock Control == Checked*. The IP core sets this to low when PHY clock needs to be updated. The interrupt status PHY Clock Update is also asserted. You must update the PHY clock (*eclk_i*/*eclk90_i*) frequency according to the *PHY Clock Rate* register. Once the updated PHY clock is ready and stable, you must set this to *1* to proceed with the LTPI Link process.
		RO	0x0	3:1	—	Reserved
		RO	0x0	7:4	PHY Clock Rate 0	Available only if *Enable Programmable Clock Control == Checked*. Indicates the requested clock frequency rate of *eclk_i* and *eclk90_i*. 4'd0 – 25 MHz (default) 4'd1 – 50 MHz 4'd2 – 75 MHz 4'd3 – 100 MHz 4'd4 – 150 MHz 4'd5 – 200 MHz 4'd6 – 250 MHz 4'd7 – 300 MHz 4'd8 – 400 MHz 4'd9 – 600 MHz
		RO	0x0	31:8	—	Reserved
0x10	OEM Channel Tx Data 0	RW	0x0	31:0	Tx Data	This is available only if *Enable Full OEM Capabilities Type == Checked*. This contains the Custom OEM Frame data to be transmitted. You may construct your own payload format. Tx Data 0 [7:0] is the start of frame symbol (0xFC), while Tx Data 3 [31:24] is the CRC. You do not need to calculate the CRC, as the IP calculates and appends it automatically.
0x14	OEM Channel Tx Data 1	RW	0x0	31:0	Tx Data	
0x18	OEM Channel Tx Data 2	RW	0x0	31:0	Tx Data	
0x1C	OEM Channel Tx Data 3	RW	0x0	31:0	Tx Data	
0x20	OEM Channel Rx Data 0	RO	0x0	31:0	Rx Data	This is available only if *Enable Full OEM Capabilities Type == Checked*. This contains the Custom OEM Frame data received from the remote link.
0x24	OEM Channel Rx Data 1	RO	0x0	31:0	Rx Data	
0x28	OEM Channel Rx Data 2	RO	0x0	31:0	Rx Data	
0x2C	OEM Channel Rx Data 3	RO	0x0	31:0	Rx Data	

Note:

1. For different mapping based on capability type, refer to the Frame Format section.

2.5. Implementation Flow

2.5.1. Tunneling

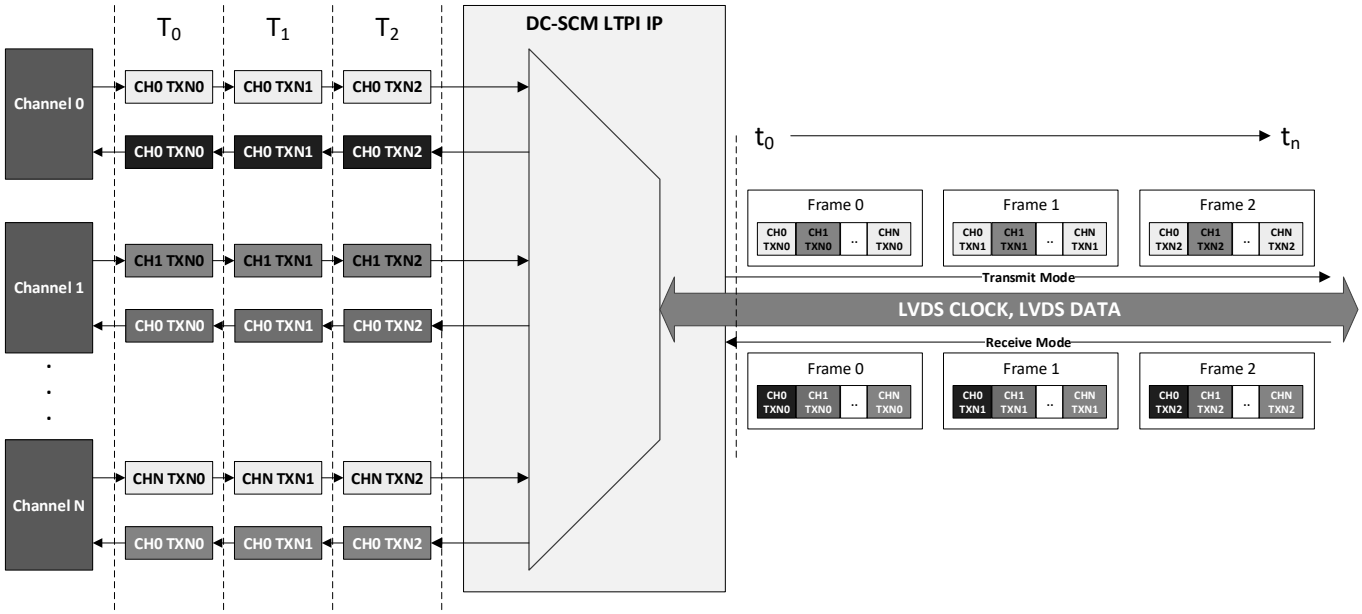


Figure 2.7. Tunneling Diagram

2.5.2. State Machine

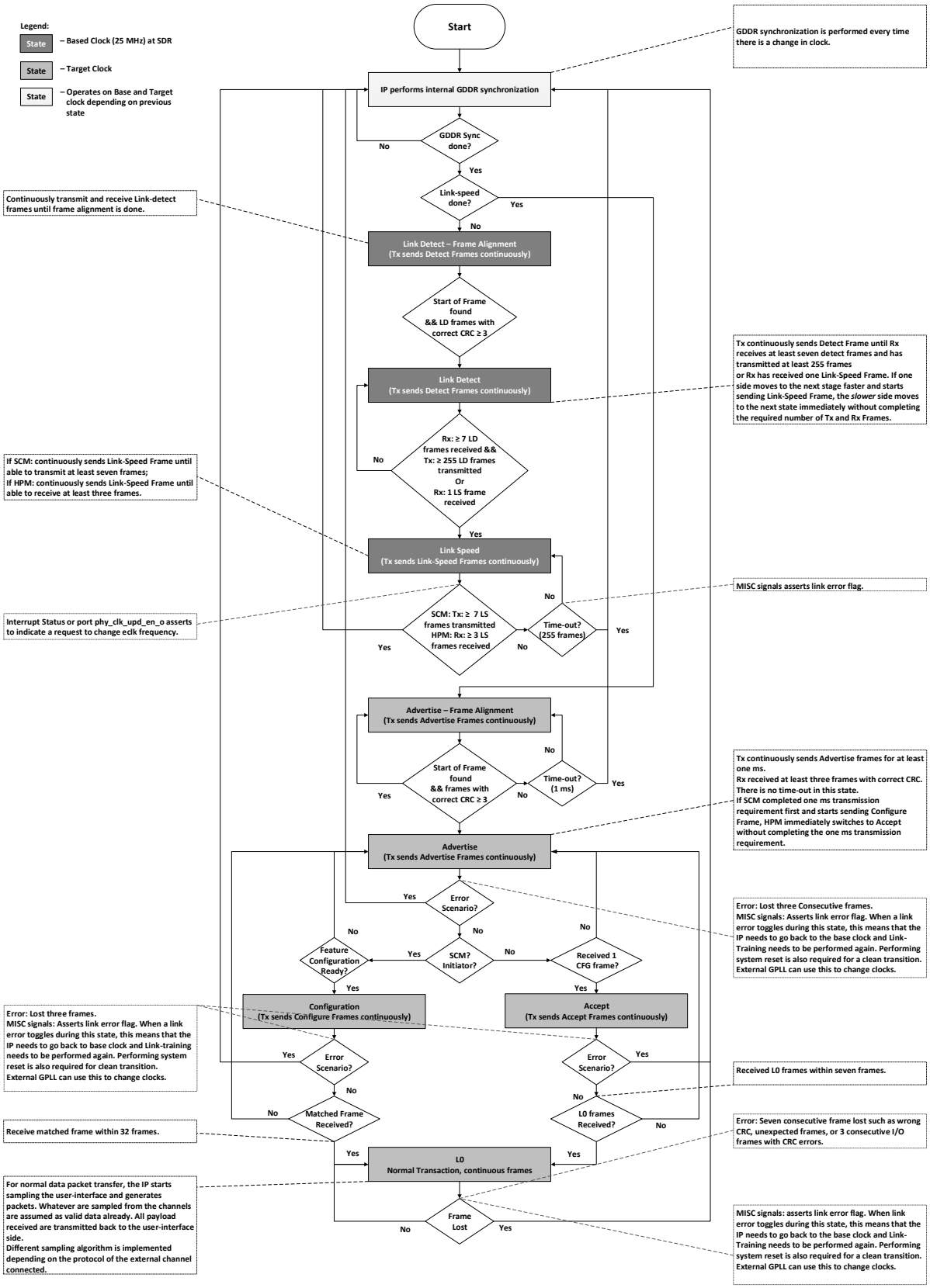


Figure 2.8. FSM of DC-SCM LTPI IP

2.5.2.1. PHY States

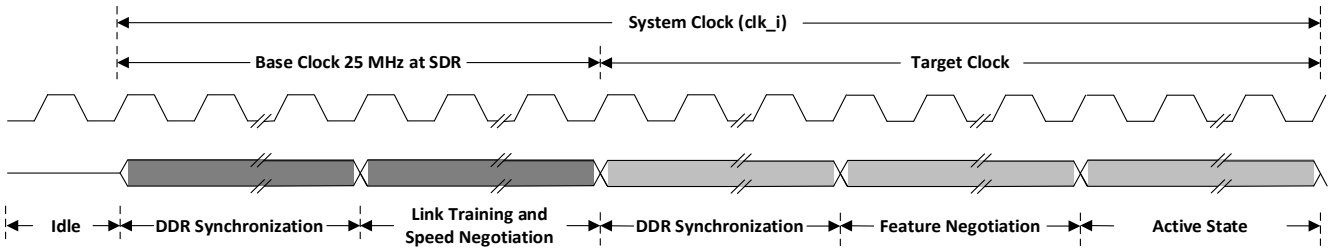


Figure 2.9. PHY States

Figure 2.9 is the high-level illustration of the states of the LTPI IP, starting from initialization up to active state:

1. DDR synchronization (GDDR Clock Synchronization State).
2. Link training and speed negotiation (Link Training – Link-Detect State and Link Training – Link-Speed State).
3. DDR synchronization.
4. Feature negotiation (Advertise State and Configuration State (for SCM only) or Accept State (for HPM only)).
5. Active state (L0 State).

2.5.2.2. GDDR Clock Synchronization State

GDDR synchronization is required by the DDR interface to properly synchronize DDR-related clocks. It automatically starts after the DDR-related clocks in the system become stable (phy_clk_valid_i=1). GDDR clock synchronization logic is always clocked by clk_i.

During initialization, the IP commences start-up by performing internal DDR synchronization. After successful GDDR synchronization, the IP proceeds to word alignment.

GDDR synchronization must be performed every time there is a change in the working frequency of the IP, that is, switching from base speed to target speed after Link Training – Link-Speed State.

2.5.2.3. Link Training – Link-Detect State

Link-Detect state is divided into two subparts:

- Link-Detect – Frame Alignment
  After initial DDR clock synchronization, the IP goes to Link-Detect – Frame Alignment State. This state is used to broadcast the speed capability of the links by sending Link-Detect Frames. Content of Link-Detect Frame is shown in Table 2.23. Tx continuously transmits Link-Detect Frame while the Rx continuously receives the frames. The Rx uses the received Link-Detect frames to do word alignment and start of frame detection. When word alignment is done and start of frame is found, the state transitions to Link-Detect after three frames with correct CRC are received. Successful alignment is indicated by the assertion of the LTPI Link Align status register. A sample timing diagram is shown in Figure 2.10.
- Link-Detect
  Frame counter is started after frame alignment is done. When the Rx has already received at least seven consecutive Link-Detect Frames and Tx has transmitted at least 255 Link-Detect Frames, the IP transitions to the next state, Link Training - Link-Speed State. If one side moves to the next stage faster and already starts sending Link Speed Frame, the slower side shall move to next state immediately without completing required number of Tx and Rx frames. SCM and HPM can transition to the Link-Speed State asynchronously.

2.5.2.4. Link Training – Link-Speed State

Link-Speed State is used to define the target operating speed of SCM and HPM. The target speed is determined based on the fastest clock capability through Link-Detect Frames in Link-Detect State that is common between SCM and HPM. During this state, Link-Speed Frame is continuously transmitted across SCM and HPM. The content of a Link-Speed Frame is shown in Table 2.24.

To transit to Advertise State from Link-Speed State, following conditions must be satisfied:

- If SCM: Tx has already transmitted at least seven Link-Speed Frames.
- If HPM: Rx has already received at least three Link-Speed Frames with the same target speed set.

Successful target speed negotiation is indicated by assertion of int_o port if interrupt is enabled and phy_clk_upd_en_o port if *Enable Programmable Clock Control == Unchecked*. Sample timing diagram is shown in Figure 2.10. If after 255 Link-Speed Frames are received and no same target speed is detected in the received frames, the IP goes back to Link-Detect state and the lnk_err_o port is asserted.

2.5.2.5. Advertise State

During the Advertise state, the IP must switch to the target speed based on the Link-Speed frames broadcasted in Link Training – Link-Speed state.

The Advertise state is divided into two subparts:

- Advertise – Frame Alignment
  After DDR clock synchronization, IP goes to Advertise – Frame Alignment State. This state is used to broadcast the feature capability of the IP through Advertise Frames. Content of Advertise Frame is shown in Table 2.25. Advertise Frame is continuously transmitted between SCM and HPM for at least one ms. The Rx uses the received Advertise Frames to do word alignment and start of frame detection in operational frequency. When word alignment is done and start of frame is found, state transitions to Advertise after three frames with correct CRC are received. Successful alignment is indicated by the assertion of the LTPI Link Aligned status register. Sample timing diagram is shown in Figure 2.10.
- Advertise
  This is the main part of Advertise State where Frames are used to interpret the LTPI capabilities of the remote side.

Both SCM and HPM shall keep sending Advertise Frames for at least one ms to allow the link to stabilize at the operational frequency. If three consecutive frames are lost during Advertise State, the IP goes back to GDDR Synchronization State and the lnk_err_o port is asserted. Initialization and Link-Training must be performed again.

Successful transition from base speed to target speed is indicated by the assertion of the phy_clk_valid_i port followed by de-assertion of phy_clk_upd_en_o if *Enable Programmable Clock Control == Unchecked*.

If at least three consecutive Advertise Frames are received and the Tx has already transmitted Advertise Frames for at least one ms, depending on the *IP Mode* set in the attribute, IP transitions to different states. If IP is set as SCM, the IP goes to Configuration State when Feature Configuration is ready. Refer to Request Features section for details on how to set the feature configuration request to move the IP to Configuration State. If IP is set as HPM, the IP goes to Accept State when it receives at least one Configuration Frame from SCM. If SCM completed one ms transmission requirement first and already starts sending Configure Frame, HPM shall immediately switch to Accept without completing the one ms transmission requirement.

In cases when the IP goes back to Advertise State from either Configuration or Accept State due to feature mismatch, you must redo feature configuration request sequence in order to correctly move the IP to Configuration State.

2.5.2.6. Configuration State (for SCM only)

During this state, SCM transmits Configure Frames. Configure Frame indicates the requested features and frame content shown in Table 2.32. If a matched Accept Frame from HPM is received, the IP goes to L0 State. If within 32 frames and no matching Accept Frame is received, the IP goes back to Advertise State. A matched frame is equivalent to Accept Frame from HPM with the same enabled features as Configure Frame sent by SCM. Request feature bits are equal to accept feature bits. If three frames are lost, the IP goes back to GDDR Synchronization State and the lnk_err_o port is asserted. Initialization and Link-Training must be performed again.

2.5.2.7. Accept State (for HPM only)

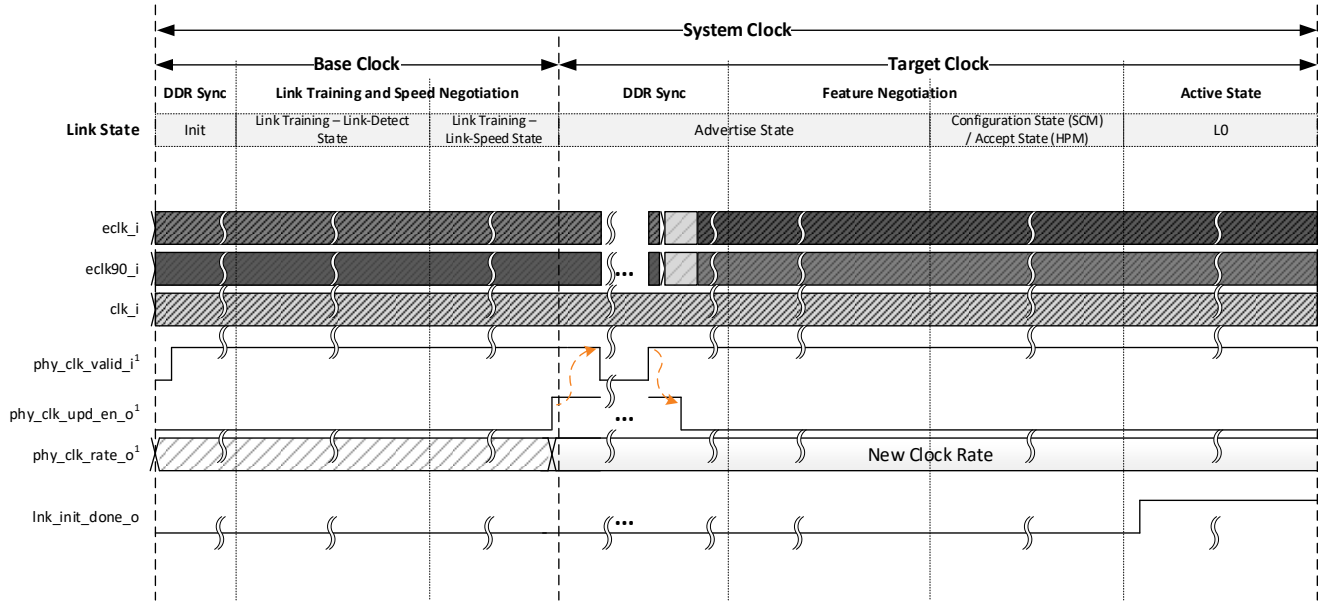
During this state, HPM broadcasts the accepted features from the received Configure Frame from SCM. Accept Frame is the AND logic of feature capability (Advertise Frame) and requested features (Configure Frame) and the frame content is shown in Table 2.33, with the exception of UART feature field which is based on highest baud rate supported. The IP continuously transmits Accept Frames to SCM until it receives L0 Frames. If L0 Frames are not received within 15 frames, the IP goes back to Advertise State. If three frames are lost, the IP goes back to GDDR Synchronization State and the lnk_err_o port is asserted. Initialization and Link-Training must be performed again.

2.5.2.8. L0 State

Normal operation happens in L0 State. Data from external channels are sampled during this state. If three consecutive frames are lost or system reset is detected, the IP goes back to DDR Synchronization State and initialization and Link-Training must be performed again. If frame lost is detected, the lnk_err_o port is asserted. If Link Software Reset (Link Control [0]) is detected, the IP goes back to Advertise State.

2.5.2.9. Timing Diagram

Figure 2.10 shows the sample LTPI Initialization to Active State timing diagram.



Note:
1. Available only if *Enable Programmable Clock Control == Checked.*

Figure 2.10. Sample LTPI Initialization to Active State Timing Diagram

2.6. Frame Format

2.6.1. Frame Format

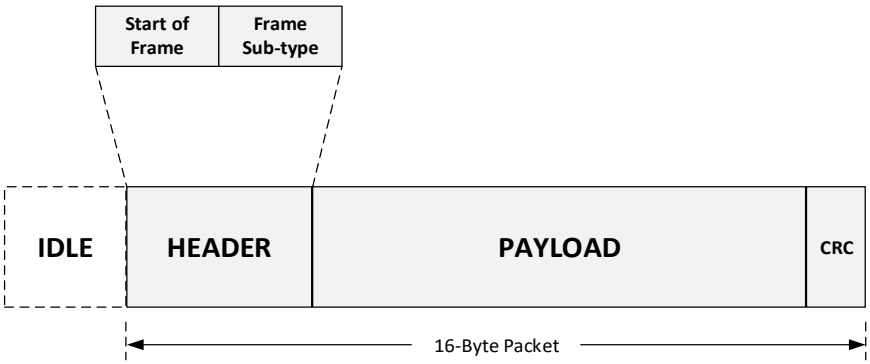


Figure 2.11. Frame Illustration

Table 2.21. Frame Format Contents

Frame Offset [Byte Number]	Bus Width (in bits)	Description
0	8	Start of Frame (SYMBOL) Indicates the start of frame.
1	8	Frame Sub-Type
2–14	13×8	Data
15	8	CRC-8 Footer that indicates the end of frame.

Table 2.21 illustrates the frame format of LTPI frame. First byte indicates the start of frame followed by an 8-bit symbol that indicates the frame sub-type. Both values depend on the current link state of the IP and the type of frame being processed, that is Link-Training Frames during Link Training and Negotiation states, and Data frames during Active State. Frame sub-type is followed by the 13-byte frame data or payload and the 8-bit CRC byte of the frame.

Table 2.22. Summary of Start of Frame and Frame Sub-type

Item	Link Training and Negotiation States				Active State
Byte	Link Training – Link Detect State	Link Training – Link Speed State	Advertise State	Configuration/Accept State	Active State
Start of Frame	K28.5		K28.6		K28.7
Frame Sub-type	D0.0	D1.0	D0.0	D1.0 (SCM) D2.0 (HPM)	0x00 (Default I/O) 0x10 (Custom I/O) 0x01 (Data)

2.6.2. Different Frame Formats

2.6.2.1. Link-Detect Frame

Link-Detect Frames are transmitted during Link Training – Link-Detect State. They consist of LTPI version and speed capability of the LTPI IP, and are later on used to determine the common target speed between SCM and HPM.

Table 2.23. Link-Detect Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.5	N/A
1	D0.0	N/A
2	LTPI Version[1,3]	Detect Capabilities Local [7:0]
3	Speed Capability 1[2,3]	Detect Capabilities Local [15:8]
4	Speed Capability 2[2,3]	Detect Capabilities Local [23:16]
5	Reserved	N/A
6	Reserved	N/A
7	Reserved	N/A
8	Reserved	N/A
9	Reserved	N/A
10	Reserved	N/A
11	Reserved	N/A
12	Reserved	N/A
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

Notes:

1. Value is from the information set for attributes *LTPI Version* during IP generation. Refer to Figure 2.12 for details on bit mapping.
2. Value is from the information set for attributes under *Speed Capability* during IP generation. Refer to Figure 2.13 for details on bit mapping.
3. Refer to the Attribute Summary section for details on the configurable IP settings.

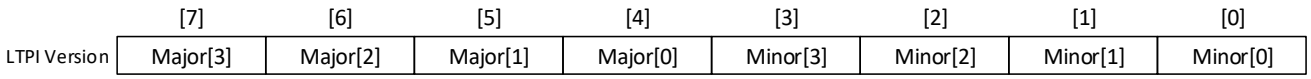


Figure 2.12. Detect Capabilities Local [7:0] Byte Mapping

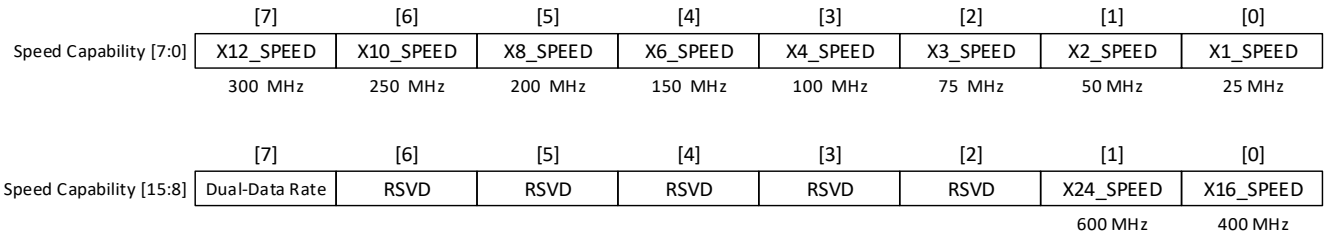


Figure 2.13. Detect Capabilities Local [23:8] Byte Mapping

2.6.2.2. Link-Speed Frame

Link-Speed Frames are transmitted during Link Training – Link-Speed state. They consist of LTPI version and target speed of the LTPI link.

Table 2.24. Link-Speed Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.5	N/A
1	D1.0	N/A
2	LTPI Version	Detect Capabilities Local [7:0]
3	Target Speed 1	N/A Processed by the IP
4	Target Speed 2	

Byte Sequence	Symbol	Corresponding Register
5	Reserved	N/A
6	Reserved	N/A
7	Reserved	N/A
8	Reserved	N/A
9	Reserved	N/A
10	Reserved	N/A
11	Reserved	N/A
12	Reserved	N/A
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

The target speed is generated from the speed capability information in Link-Detect frames. It is determined based on the fastest clock capability that is common between SCM and HPM. In the example in Figure 2.14, speed capability of SCM Link-Detect frames supports 200 MHz, 100 MHz, and 25 MHz DDR clocks while HPM supports 100 MHz and 25 MHz DDR clocks. The fastest common speed between the two is 100 MHz DDR. The IP uses this as the target speed for Link-Speed Frame.

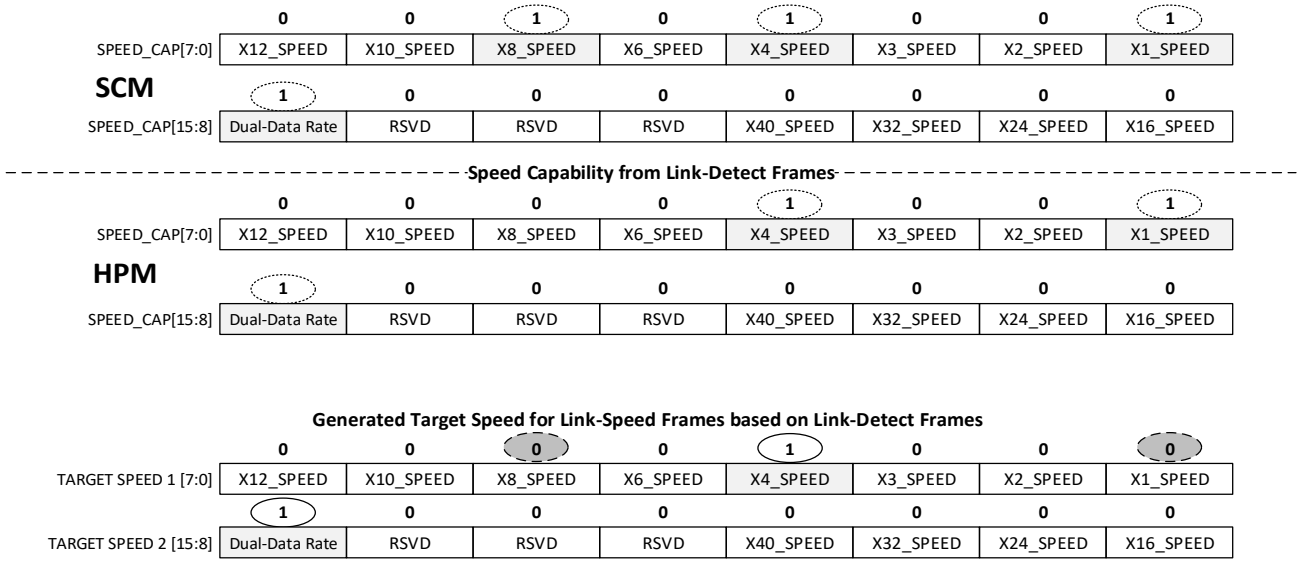


Figure 2.14. Target Speed Processing Illustration

2.6.2.3. Advertise Frame

Advertise Frames are transmitted during Advertise State. Advertise Frame consists of the feature capabilities of SCM and HPM.

Table 2.25. Advertise Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	N/A
1	D0.0	N/A
2	Platform Type Byte 0[1,4]	Platform ID Local [7: 0]
3	Platform Type Byte 1[1,4]	Platform ID Local [15: 8]
4	*Capabilities Type*[2,4]	N/A
5	Feature Capability 1[3,4]	Advertise Capabilities Local Low [7: 0]
6	Feature Capability 2[3,4]	Advertise Capabilities Local Low [15: 8]
7	Feature Capability 3[3,4]	Advertise Capabilities Local Low [23:16]

Byte Sequence	Symbol	Corresponding Register
8	Feature Capability 4[3,4]	Advertise Capabilities Local Low [31:24]
9	Feature Capability 5[3,4]	Advertise Capabilities Local High [7: 0]
10	Feature Capability 6[3,4]	Advertise Capabilities Local High [15: 8]
11	Feature Capability 7[3,4]	Advertise Capabilities Local High [23:16]
12	Feature Capability 8[3,4]	Advertise Capabilities Local High [31:24]
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

Notes:

1. Value is from the information set for attribute *Platform Type ID (16b Hex)* during IP generation. Refer to Table 2.26 for details on bit mapping.
2. Value is from the information available for attribute *Capabilities Type* during IP generation. Refer to Table 2.27 for more details.
3. Feature capability varies depending on *Capabilities Type* set. Refer to Table 2.28, Table 2.29, and Table 2.30 for details.
4. Refer to Attribute Summary section for details on the configurable IP settings.

Table 2.26. Platform Field Bit Mapping

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	ID (16-bit ID Number set in Attributes)							
1								

Table 2.27. Capabilities Type Details

Capabilities Type	Description
0x00	Used for default Feature Capability mapping defined in LTPI Specifications as shown in Table 2.28.
0x01-0x80	LTPI Reserved.
0x81	Used for predefined Feature Capability mapping defined by the IP for custom type as shown in Table 2.29.
0x82–0xFF	OEM defined When defined, Feature Capability mapping is as shown in Table 2.30.

Feature Capability mapping is dependent on the Capabilities Type selected. Table 2.28 is the mapping of feature capability predefined in DC-SCM LTPI standard.

Table 2.28. Feature Capability Mapping for Default I/O Frame (*Capabilities Type*==0x00)

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	External Channel Enable							
	RSVD			OEM	Data Channel	UART	I2C	GPIO
1	Number of NL GPIO[7:0][1]							
2	RSVD						Number of NL GPIO[9:8]	
3	RSVD	Echo Support[2] 0:Disabled 1: Enabled	I2C BUS5[3] 0: Disabled 1: Enabled	I2C BUS4[3] 0: Disabled 1: Enabled	I2C BUS3[3] 0: Disabled 1: Enabled	I2C BUS2[3] 0: Disabled 1: Enabled	I2C BUS1[3] 0: Disabled 1: Enabled	I2C BUS0[3] 0: Disabled 1: Enabled
4	I2C Channel Capabilities							
	RSVD		I2C BUS5 SP 0: 100 kHz 1: 400 kHz	I2C BUS4 SP 0: 100 kHz 1: 400 kHz	I2C BUS3 SP 0: 100 kHz 1: 400 kHz	I2C BUS2 SP 0: 100 kHz 1: 400 kHz	I2C BUS1 SP 0: 100 kHz 1: 400 kHz	I2C BUS0 SP 0: 100 kHz 1: 400 kHz
5	UART Channel Capabilities							

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
	RSVD	UART1[3] 0: Disabled 1: Enabled	UART0[3] 0: Disabled 1: Enabled	Flow Control	*Baud Rate* Encoding (refer to Table 2.31)			
6	OEM Capability [7:0][4]							
7	OEM Capability [15:8][4]							

Notes:

1. Total Number of NL GPIO that is tunneled over LTPIO, the *NL GPIO Input Data Width* (or the *NL GPIO Output Data Width*, if it is higher) attribute.
2. Echo support is always enabled for the I2C channel.
3. Unused local bus interface must be driven to HIGH if bus is initially enabled during IP configuration but is disabled during final LTPI feature configuration states.
4. Value is from the information set for attributes *OEM Capability (16b Hex)* during IP generation.

Table 2.29 shows the mapping of feature capability defined by the IP for Custom I/O frame.

Table 2.29. Feature Capability Mapping for Custom I/O Frame (*Capabilities Type*==0x81)

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	External Channel Enable							
	RSVD		OEM	Data Channel	UART	I2C	NL GPIO	LL GPIO
1	Number of NL GPIO[7:0][1]							
2	RSVD						Number of NL GPIO[9:8]	
3	UART Channel Capabilities							
	RSVD			Flow Control	*Baud Rate* Encoding (refer to Table 2.31)			
4	I2C Channel Capabilities[2]							
	I2C BUS7 SP 0: 100 kHz 1: 400 kHz	I2C BUS6 SP 0: 100 kHz 1: 400 kHz	I2C BUS5 SP 0: 100 kHz 1: 400 kHz	I2C BUS4 SP 0: 100 kHz 1: 400 kHz	I2C BUS3 SP 0: 100 kHz 1: 400 kHz	I2C BUS2 SP 0: 100 kHz 1: 400 kHz	I2C BUS1 SP 0: 100 kHz 1: 400 kHz	I2C BUS0 SP 0: 100 kHz 1: 400 kHz
5	Additional I2C Channel Capabilities							
	I2C BUS15 SP 0: 100 kHz 1: 400 kHz	I2C BUS14 SP 0: 100 kHz 1: 400 kHz	I2C BUS13 SP 0: 100 kHz 1: 400 kHz	I2C BUS12 SP 0: 100 kHz 1: 400 kHz	I2C BUS11 SP 0: 100 kHz 1: 400 kHz	I2C BUS10 SP 0: 100 kHz 1: 400 kHz	I2C BUS9 SP 0: 100 kHz 1: 400 kHz	I2C BUS8 SP 0: 100 kHz 1: 400 kHz
6	Additional I2C Channel Capabilities							
	I2C BUS23 SP 0: 100 kHz 1: 400 kHz	I2C BUS22 SP 0: 100 kHz 1: 400 kHz	I2C BUS21 SP 0: 100 kHz 1: 400 kHz	I2C BUS20 SP 0: 100 kHz 1: 400 kHz	I2C BUS19 SP 0: 100 kHz 1: 400 kHz	I2C BUS18 SP 0: 100 kHz 1: 400 kHz	I2C BUS17 SP 0: 100 kHz 1: 400 kHz	I2C BUS16 SP 0: 100 kHz 1: 400 kHz
7	OEM Capability [7:0][3]							

Notes:

1. Total Number of NL GPIO that is tunneled over LTPIO, the *NL GPIO Input Data Width* (or the *NL GPIO Output Data Width*, if it is higher) attribute.
2. Echo support is always enabled for the I2C channel.
3. Value is from the information set for attribute *OEM Capability (16b Hex)* during IP generation.

When *Capabilities Type == 0x82-0xFF* (OEM defined) is used, feature capability is solely based on user-defined value and mapping is shown in Table 2.30. Definition of actual feature mapping for OEM defined capability type is out of scope of the soft IP. For the external channels, the IP uses the default feature settings set during the IP generation. For example, I2C channel follows the I2C bus speed set during IP generation, 100 kHz if *Enable Fast Mode == 0* and 400 kHz if *Enable Fast Mode == 1*.

Table 2.30. Feature Capability Mapping for OEM Defined (*Capabilities Type == 0x82-0xFF*)

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	OEM Feature Capability 0 in Hex[1]							
1								
2								
3								
4	OEM Feature Capability 1 in Hex[1]							
5								
6								
7								

Note:

1. Value is from the information set for attributes *OEM Feature Capability (16b Hex)* during IP generation.

Table 2.31. UART Baud Rate Encoding

Baud Rate (bits per second)	4-bit Encoding
300	0x0
600	0x1
1200	0x2
1800	0x3
2400	0x4
4800	0x5
9600	0x6
19200	0x7
38400	0x8
57600	0x9
115200	0xa
230400	0xb
460800	0xc
576000	0xd
921600	0xe
RSVD	0xf

2.6.2.4. Configure Frame

Configure Frames are transmitted by SCM during Configuration State. Configure Frame consists of the requested features set by SCM based on feature capability information from Advertise Frame. Request Feature fields must follow the same bit mapping as Feature Capability field of Advertise Frame. Refer to the Request Features section for more details on feature request.

Table 2.32. Configure Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	N/A
1	D1.0	N/A
2	Capabilities Type	N/A
3	Request Feature 1[1]	Advertise Capabilities Local Low [7: 0]
4	Request Feature 2[1]	Advertise Capabilities Local Low [15: 8]
5	Request Feature 3[1]	Advertise Capabilities Local Low [23:16]
6	Request Feature 4[1]	Advertise Capabilities Local Low [31:24]
7	Request Feature 5[1]	Advertise Capabilities Local High [7: 0]
8	Request Feature 6[1]	Advertise Capabilities Local High [15: 8]
9	Request Feature 7[1]	Advertise Capabilities Local High [23:16]

Byte Sequence	Symbol	Corresponding Register
10	Request Feature 8[1]	Advertise Capabilities Local High [31:24]
11	Reserved	N/A
12	Reserved	N/A
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

Note:

1. Must follow the same bit mapping as Feature Capability field of Advertise Frame.

2.6.2.5. Accept Frame

Accept Frames are transmitted by HPM during Accept State. Accept Frame indicates which features are accepted by the HPM based on the received Configure Frames from SCM.

Table 2.33. Accept Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	N/A
1	D2.0	N/A
2	Capabilities Type	N/A
3	Accepted Feature 1	N/A Processed by the IP
4	Accepted Feature 2	
5	Accepted Feature 3	
6	Accepted Feature 4	
7	Accepted Feature 5	
8	Accepted Feature 6	
9	Accepted Feature 7	
10	Accepted Feature 8	
11	Reserved	N/A
12	Reserved	N/A
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

Accepted features are the AND logic of Feature Capability fields from Advertise Frame of HPM and Request Feature fields of Configure Frame from SCM with the exception of UART feature related field. For UART field, requested baud rate is accepted by HPM if it is less than or equal to the HPM UART baud rate set in the Feature Capability field.

If *Capabilities Type* is OEM defined, Accept Frame is the AND logic of all fields of Feature Capability (Advertise Frame) of HPM and Request Feature (Configure Frame) from SCM.

2.6.2.6. Data/I/O Frames

Data/I/O frames are transmitted during L0 state. Data/I/O frame consists of the actual payload from external channels. Mapping of payload in a frame depends on which *Data Frame Type* is selected. Summary of interface bus width support is shown in Table 2.34.

Table 2.34. Data Frame Type Summary

Category	Size	Custom I/O Frame	Default I/O Frame	Data Frame
Frame Sub-Type	bits	0x10	0x00	0x01
LL GPIO Channel	bits	User setting based on Attributes	16 bits	16 bits
I2C Channel	1 bus = 4 bits		6 buses	N/A
OEM Channel	bits		32 bits	N/A
UART Channel	1 bus = 4 bits		2 buses	N/A
NL GPIO Channel	bits		16 bits	N/A
NL Frame Counter	bits	Based on NL GPIO settings		N/A
DATA Channel	bits	N/A	N/A	88

Custom I/O Frame

Custom I/O frame is a basic I/O frame format in which frame field arrangement for enabled channels are user-customizable. The Tx generates a 13-byte wide payload with active data length varying upon the payload size set per frame per channel. The payload has 12-byte maximum active data and 1-byte for NL Frame Counter. If the user-set payload length is less than 13-bytes, payload is padded with 0s to complete the 13-byte payload.

The channel sequence mapping in the payload field is customizable in the Frame Format tab in the attribute section during IP generation. Refer to Table 2.4 for details. If any of the channels is disabled, it is not mapped to any of the payload field available. Payload byte offset allocation for each enabled channel is indicated by the *<Channel> (Start Index)* attribute. The number of bytes assigned for each enabled channel is indicated by *<Channel>* (*Byte Size)* set attribute.

The Frame Format tab in the GUI is shown in Figure 2.15 and the corresponding payload mapping in shown in

In Table 2.35. In Table 2.35, the sample Custom I/O frame has user-custom payload setting of two bytes LL GPIO, two bytes NL GPIO, six I2C buses, two UART buses, and four bytes OEM with channel sequence mapping of LL GPIO, NL GPIO, I2C, UART, OEM, and NL Frame Counter. You have the option to rearrange the sequence mapping of the enabled channels during IP generation.

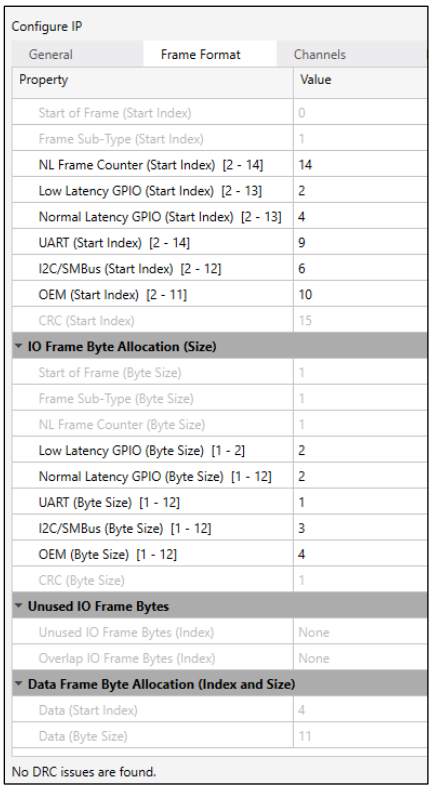


Figure 2.15. Sample Channel Order for Custom I/O

Table 2.35. Sample Custom I/O Format

Byte Sequence	Symbol	Corresponding Register
0	K28.7	N/A
1	0x10	N/A
2	LL GPIO 01	N/A Processed by the IP
3	LL GPIO 02	
4	NL GPIO 01	
5	NL GPIO 02	
6	I2C Bus 1 and 0	
7	I2C Bus 3 and 2	
8	I2C Bus 5 and 4	
9	UART Bus 1 and 0	
10	OEM Reserved 0	
11	OEM Reserved 1	
12	OEM Reserved 2	
13	OEM Reserved 3	
14	NL Frame Counter	
15	CRC	N/A (Computed by the IP)

Default I/O Frame

Default I/O frame is also a basic I/O frame format but channel allocation in frame is predefined based on DC-SCM LTPI standard. This is used to aggregate normal and low latency GPIOs, I2C, UART, and OEM with fixed number of channels and frame allocation bytes. NL Frame Counter always occupies the third byte of Default I/O frame. Payload byte offset allocation for each channel is indicated by the *<Channel> (Start Index)* attribute. If any of the channels is disabled, the corresponding allocated payload field is padded with 1s. You cannot rearrange the sequence mapping of the enabled channels during IP generation.

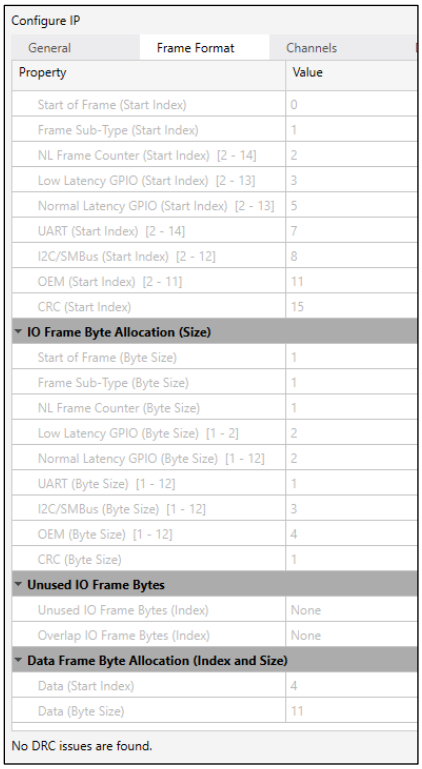


Figure 2.16. Frame Format Tab for Default I/O Frame

Table 2.36. Default I/O Format

Byte Sequence	Symbol	Corresponding Register
0	K28.7	SP_SYMBOL
1	0x00	DFT_SYMBOL
2	NL Frame Counter	N/A Processed by the IP
3	LL GPIO 01	
4	LL GPIO 02	
5	NL GPIO 01	
6	NL GPIO 02	
7	UART Bus 1 and 0	
8	I2C Bus 1 and 0	
9	I2C Bus 3 and 2	
10	I2C Bus 5 and 4	
11	OEM Reserved 0	
12	OEM Reserved 1	
13	OEM Reserved 2	
14	OEM Reserved 3	
15	CRC	N/A (Computed by the IP)

Data Frame

Data Frame is a memory-type frame format used when Data Channel is enabled. This is considered as a Random-Access frame and is only sent on demand when there is a Data Write/Read request. In this *Data Frame Type*, only low-latency GPIO and Data Channel are processed. Payload byte allocation is predefined. Refer to Data Channel section for the actual command mapping. For cases when *Enable Low Latency GPIO Channel == Unchecked*, LL GPIO 01/02 are set to high by default.

Table 2.37. Data Frame General Format

Byte Sequence	Symbol	Corresponding Register
0	K28.7	SP_SYMBOL
1	0x01	N/A
2	LL GPIO 01	N/A Processed by the IP
3	LL GPIO 02	
4	data_ch[7:0]	
5	data_ch[15: 8]	
6	data_ch[23:16]	
7	data_ch[31:24]	
8	data_ch[39:32]	
9	data_ch[47:40]	
10	data_ch[55:48]	
11	data_ch[63:56]	
12	data_ch[71:64]	
13	data_ch[79:72]	
14	data_ch[87:80]	
15	CRC	N/A (Computed by the IP)

2.6.3. CRC

The information stored in the last byte of a valid frame is an eight order CRC code (CRC-8). This is used to detect transfer errors in the payload. The CRC is calculated for the entire data in the LTPI frame after the Start of Fame/Comma Symbol, including the frame sub-type and the payload.

The IP computes the payload CRC-8 using the polynomial shown below:

- Polynomial – x8 + x2 + x1 + 1
- CRC-8 Polynomial in Hex – 8'h07
- CRC-8 Initial Value – 8'h00

CRC is computed without inverted/reflected input and output, that is in the stream of bytes, the most significant bit of each byte is processed first to get the final CRC value.

2.6.4. Frame Interleave

When Data Channel is disabled, the LTPI frame transfer is just a continuous stream of I/O frames.

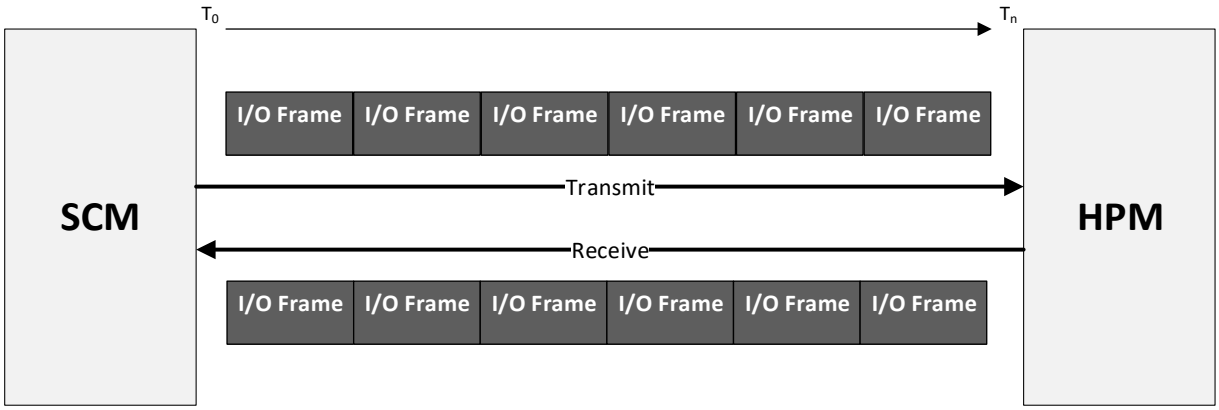


Figure 2.17. LTPI Frame Stream without Data Channel

When Data Channel is enabled, the LTPI frame transfers an interleaved transfer of I/O and Data frames, depending on when Data Frames are available.

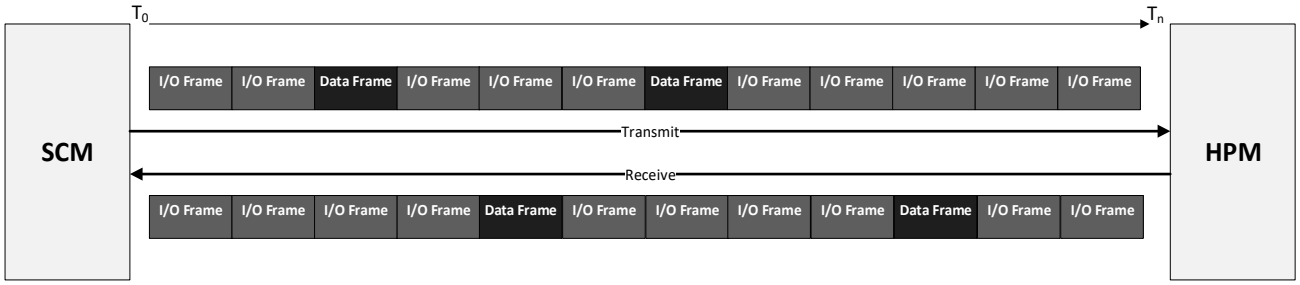


Figure 2.18. LTPI Frame Interleave with Data Channel

2.7. Functional Blocks

2.7.1. Multiplexor

Multiplexor interfaces with the external channel. After link training and feature negotiation, IP samples data from external channels. All sampled data are considered valid data. Module switches sampling between each channel to form the payload. In cases when the sampled ports are incomplete, the IP pads 0s to complete the channel payload.

Figure 2.19 shows the sample waveform. Assuming each channel has one byte allocation in the frame payload, and synchronous sampling of payload happened at positive edge of t_1, the resulting frame is shown in Figure 2.20.

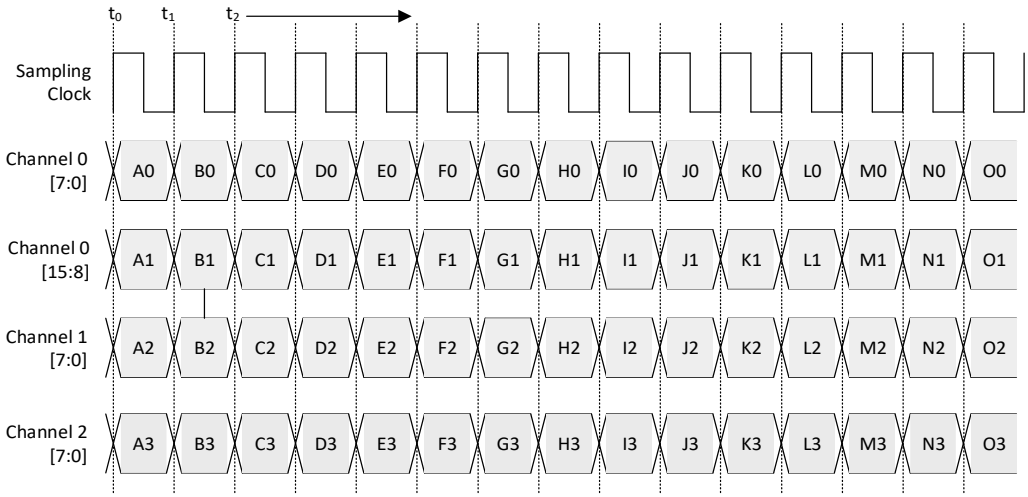


Figure 2.19. Sample Waveform

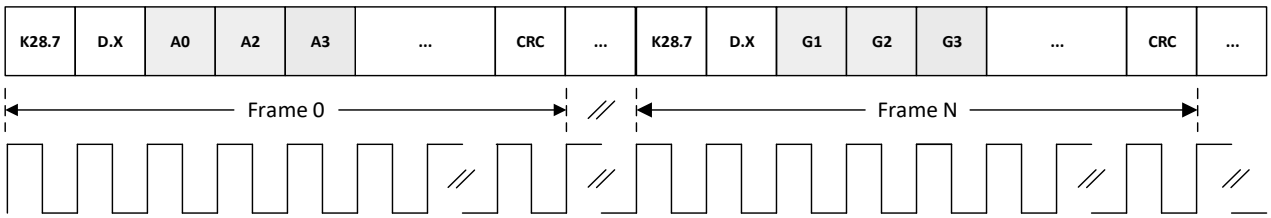


Figure 2.20. Resulting Frame

Depending on the protocol of the external channel connected, different sampling and remapping algorithms are implemented for frame generation/parsing. Refer to External Channel Interface Handling for more details.

2.7.2. Frame Generator and Parser

Frame generator and parser generates and recovers the packets for LTPI transfer. Frame generator is used by the Tx to generate the frames to be sent over to communicating receiver. Frame parser is used by Rx to parse the received frame. General content of a frame is shown in Figure 2.11. One complete frame consists of header, payload, and footer. Actual content of the frame depends on the type of frame being generated/parsed. Refer to the Frame Format section for details.

2.7.3. 8b/10b Encoder/Decoder

The IP performs 8b/10b encoding/decoding for data transmitted/received to/from remote IP. For the Tx, an 8-bit data is converted to a 10-bit data based on encoding specified in IEEE Standard 802.3, Table 36–1a–e and Table 36–2 before it is transmitted. For the Rx, the 10-bit data received from the host is decoded and converted back to 8-bit format.

2.7.4. Serializer/Deserializer

Communication between SCM and HPM in LTPI is in serial form. The IP serializes the data through generic DDR interface. Likewise for the Rx mode, data is de-serialized through the DDR interface. DDR clock and data follow center-alignment (90 degrees phase shift with respect to each other). Dynamic switching from LVDS SDR to DDR mode is implemented through emulating SDR by sending the same data in both edge of LVDS clock.

Parallel to serial conversion and vice versa follows little endian scheme. The least significant bit of the parallel data is transmitted first in LVDS data bus.

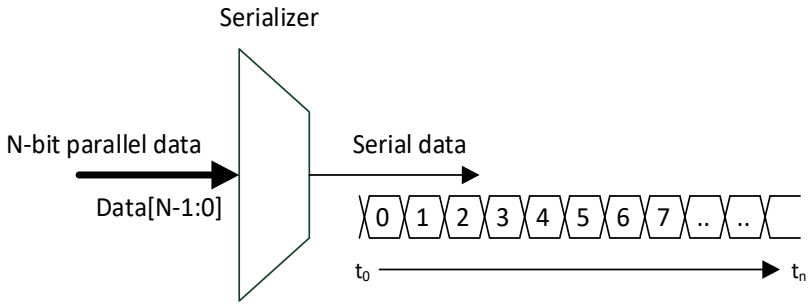


Figure 2.21. Parallel to Serial Conversion Order

2.7.5. Word Aligner

Word aligner is used by the GDDR Rx to align the incoming data. After GDDR synchronization, word aligner continuously rotates the received parallel data by asserting ALIGNWD port of GDDR Rx module until it can get the correct word alignment. This ALIGNWD port is used by GDDR to rotate the data by one bit.

Depending on the current state of the IP, different patterns are used by the word aligner. During Link-Training, the K28.5 symbol is used, while the K28.6 symbol is used during Advertise State.

2.7.6. CSR

The APB interface is used to access the IP CSR module. This CSR module contains different registers that can be used to configure the IP. For a complete list of supported registers, refer to Table 2.20.

3. IP Programming Sequence

Clock switching information is available in both CSR and IP ports. You can opt to get information from either the IP ports or from the CSR when *Enable Programmable Clock control == Checked*.

Request feature information is available in CSR and can be read after the clock switching is done and the LTPI Link Aligned status register is asserted.

3.1. Clock Switching

During link-training and negotiation, SCM and HPM transmit their corresponding speed capability, which the IP parses to generate the target speed. Target speed information is available in the CSR and IP ports. See the Timing Diagram section for the clock control timing diagram.
If *Enable Programmable Clock control* is selected, the register equivalent of the clock control interface is implemented.

When the IP is ready for clock reconfiguration, the interrupt status *PHY Clock Update* is asserted. You can poll the register or enable the interrupt and wait for int_o port to assert. During this time, the clock must be reconfigured to the target frequency based on the information read from *PHY Clock Rate* in the *Clock Control* register. Once external clock reconfiguration is done, *PHY Clock Valid* must be set to *1*. Asserting the *PHY Clock Valid* also clears the interrupt status *PHY Clock Update*.

3.2. Request Features

The feature request is primarily for *IP Mode ==* SCM. The IP is ready for feature request after clock switching is done and LTPI Link Aligned status has asserted. The *Received* feature capability information is available in registers *Advertise Capabilities Remote Low* and *High*. After reading these registers, you can use the information to determine the requested features.

By default, the *Advertise Capabilities Local Low* and *High* are set to the feature capability configured during IP generation, which is also the same value in the *Default Configuration Low* and *High* registers. If you intend to update the feature request, the *Advertise Capabilities Local* registers must be programmed accordingly.
Once the request feature-related programming is done, you must set the *Link Control[11] register Trigger* to Configuration State. This register is used by the IP to determine if the request feature-related programming is done allowing feature negotiation to resume.

If *Automatically move to Configuration State* (Link Control[10]) is set during IP configuration, the IP automatically transitions to Configuration State after completing the required frame transmission for Advertise State, without waiting for Feature related programming and Trigger to Configuration State to be set and done. You must make sure that IPs are configured in such a way that SCM and HPM have a matching feature when the attribute *Automatically move to Configuration State* is enabled. There is no way to manually reconfigure the requested features when this attribute is set.

Only when either Trigger to Configuration State register or Automatically move to Configuration State register is properly set that the IP goes to Configuration State. Trigger to Configuration State is automatically set to default value when *Link Software Reset* (Link Control [0]) is applied during normal operation state.

4. External Channel Interface Handling

During L0/Normal Operation State, the IP samples data from external channels and create Data/I/O frames. Different tunneling principles are implemented for each external channel interface.

Table 4.1. Tunneling Principles for Different Channels

Channel	Capture Method	Tx and Rx Synchronization	Channel Characteristics
GPIO	Sampling	Asynchronous	• Captured signal levels are transmitted directly through LTPI. • Low Latency GPIO are updated in every LTPI frame. • Normal Latency GPIO are split across multiple frames with the frame counter that is used to identify the NL GPIO subset.
UART			• Captured signal levels are transmitted directly through LTPI. • UART signals are oversampled. • Multiple samples are tunneled in every LTPI frame.
I2C	Event/State Detection	Synchronous	• I2C states are encoded into LTPI events and events are tunneled through LTPI. • I2C Clock stretching is used while waiting for synchronization to be completed after event is transmitted and for I2C state/event from the other side of the LTPI interface.
Data	Random Access		• Data bus transaction such as Data Read and Write are encoded into LTPI Events. • Completion indication of Bus Operation is used to synchronize access to the Data Bus.

Table 4.2 is a sample customized payload configuration that is used in this section for describing the different sampling mechanism of each external channel.

Table 4.2. Sample Customized Payload Configuration

Attribute	Selected Values
Data Frame Type	Custom
Enable Low Latency GPIO Channel	Enabled
LL GPIO Input Data Width	8
LL GPIO Output Data Width	8
LL GPIO Byte Size	2
Enable Normal Latency GPIO Channel	Enabled
NL GPIO Input Data Width	64
NL GPIO Output Data Width	64
NL GPIO Byte Size	1
Enable I2C Channel	Enabled
Total Number of I2C bus interface	6
I2C Byte Size	3
Enable UART Channel	Enabled
Number of UART bus interface	2
UART Byte Size	1
Enable OEM Channel	Enabled
OEM Data Width	16
OEM Byte Size	2
Enable Data Channel	Disabled

4.1. GPIO Channel

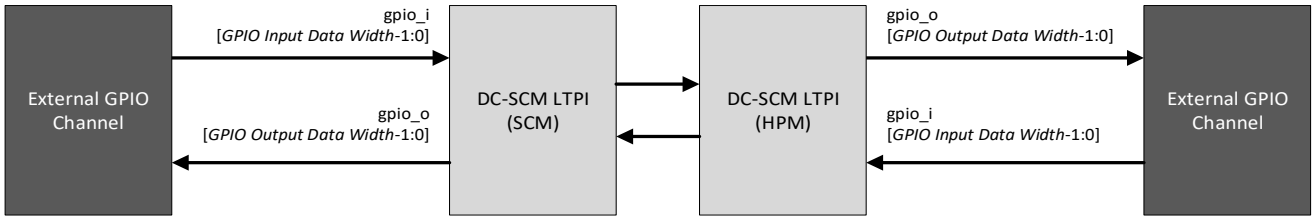


Figure 4.1. GPIO Pin List

For normal and low latency GPIO, payload allocation for both Rx and Tx frames depends on the *NL/LL GPIO Byte Size* attribute set. For low latency GPIO, each GPIO pin is directly mapped to frame and is updated every frame transfer. For normal latency GPIO, each GPIO pin update can span to multiple frame transfer.

In the LL GPIO example in Table 4.2, two bytes are allocated for LL GPIO Byte Size. If there is a total of eight *LL GPIO Input Data Width*, IP samples the 8-bit LL GPIO input pins and pad 0s to the remaining 8-bit to complete the two-byte payload allocation. For the received frames, IP remaps the received LL GPIO payload into the available GPIO output pins.

For NL GPIO, in the Tx, the IP samples the virtual GPIO based on *NL GPIO Byte Size* attribute set and multiplexes sampling around the total *NL GPIO Input Width or NL GPIO Output Width*. To maintain the same latency in both directions, a common NL GPIO Data width is selected based on the higher value between *NL GPIO Input Width* and *NL GPIO Output Width*. Index sampling is determined by the NL Frame Counter which occupies the third byte of the Default I/O frame and is customizable in the Custom I/O frame. This information is used by SCM and HPM to decode which index of NL GPIO is being tunneled in the current frame. Maximum count value (*M*) indicates the number of frames it takes to update all available NL GPIO pins. When NL GPIO is disabled, frame counter is fixed to 0. NL Frame Counter starts at count 0 that wraps around every *M-1* count based on the equation below:

$$\text{If } NL\ GPIO\ Data\ Width \geq (8 * NL\ GPIO\ Byte\ Size):$$

$$M = Ceiling(\frac{NL\ GPIO\ Data\ Width}{(8 * NL\ GPIO\ Byte\ Size)})$$

$$\text{Else:}$$

$$M = 0;$$

In the example in Table 4.2, one byte is allocated for NL GPIO payload per frame. If there is a total of 64 *Number of NL GPIO*, the IP switches sampling between each 8-bit virtual GPIO input pins per sampling time with NL Frame Counter starting at 0, until it can wrap around the total input pins with NL Frame Counter wrapping around at 7. For the received frames, IP remaps the received NL GPIO payload into the virtual GPIO output pins. Refer to Figure 4.2 for illustration. It is up to the system level design to map the virtual GPIOs to actual input and output physical ports.

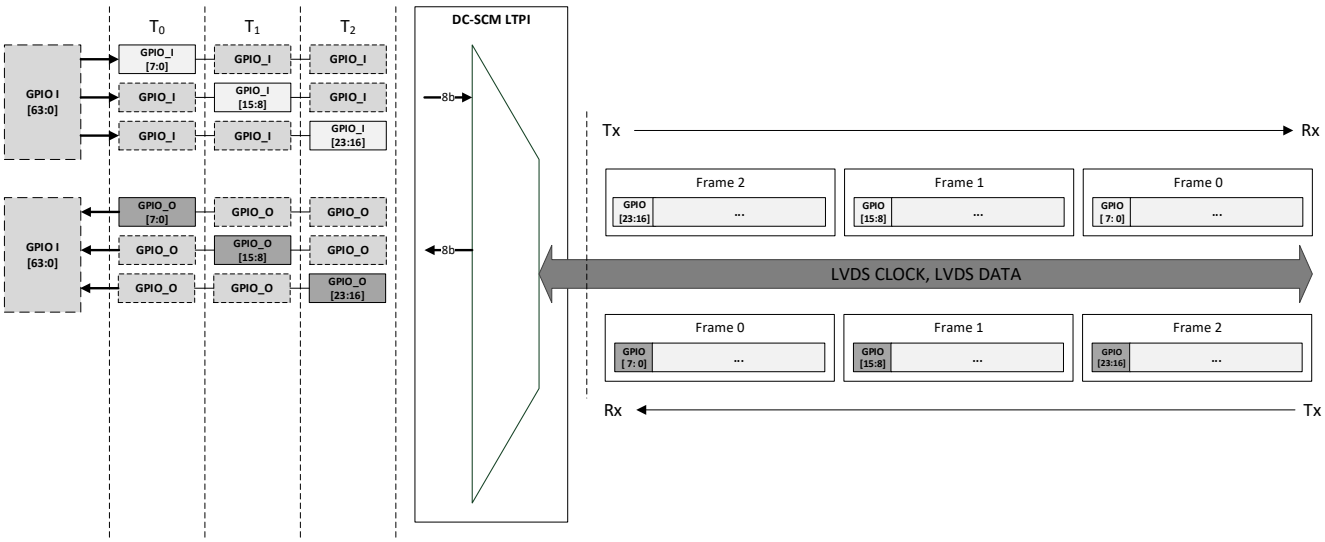


Figure 4.2. NL GPIO Mechanism

4.2. I2C Channel

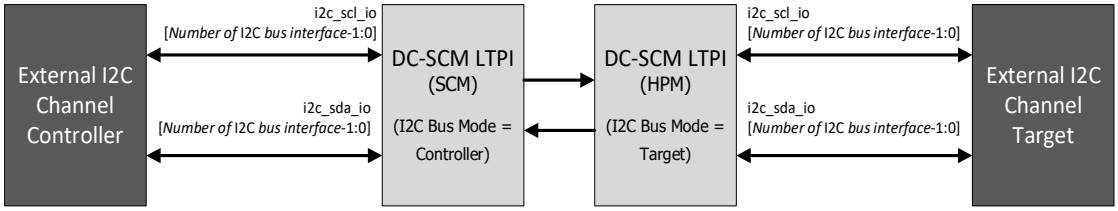


Figure 4.3. I2C Pin List

The I2C bus function can be implemented as either target, controller, or bidirectional (having both controller and target). You can set the appropriate number of interface per function as necessary. Note that the number of interfaces depends on how many bytes were allocated in the Frame. The LTPI I2C bus order (lowest index on the right) in SCM mode is {controller (external target), target (external controller), bi-directional}, while in HPM mode, {target (external controller), controller (external target), bi-directional}. Refer to the Signal Description section for more details. The same bus order is also followed in the frame payload arrangement.

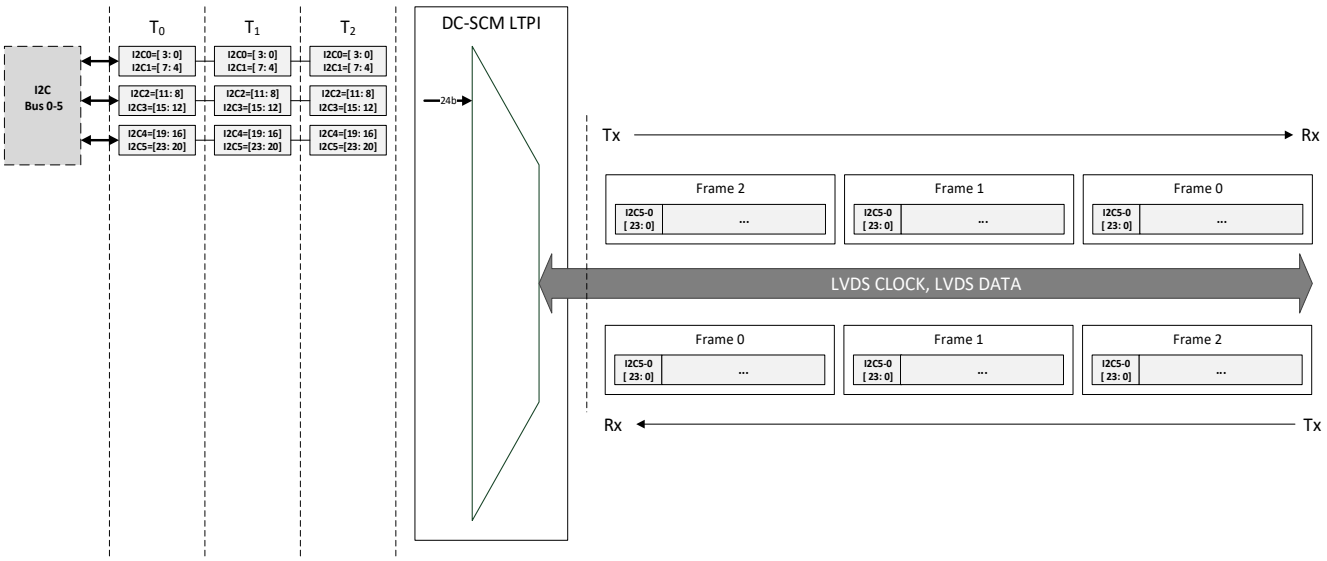


Figure 4.4. I2C Mechanism

Same with other channels, payload allocation for both Rx and Tx frames depends on the *I2C Byte Size* attribute set. When enabled, I2C bus interface is updated every non-Data Channel frame transfer. Tunneling of the I2C channel uses the Clock Stretching method defined in the I2C specifications to compensate for the LTPI latency and turnaround time. Each I2C events are encoded and captured on one side and recovered on the other side of LTPI.

Table 4.3. I2C Events Encoding

Bus Event	Starting Direction	Description	4-bit Event Encoding
Idle	Bidirectional	Idle state	0b0000
Start	Controller to Target	Start Event detected on Controller side.	0b0001
Start Received	Target to Controller	Start Event recovered on Target side.	0b0010
Stop	Controller to Target	Stop Event detected on Controller side.	0b0011
Stop Received	Target to Controller	Stop Event recovered on Target side.	0b0100
Data Received	Bidirectional	SDA bit value recovered on remote side.	0b0101
Data 0	Bidirectional	Send SDA bit value 0.	0b0110
Data 1	Bidirectional	Send SDA bit value 1.	0b0111
Start Echo	Target to Controller	Event used to indicate that Start Event was correctly received by Target side.	0b1000
Stop Echo	Target to Controller	Event used to indicate that Stop Event was correctly received by Target side.	0b1001
Data 0 Echo	Bidirectional	Send SDA bit value 0 received correctly (sent at least 3 times).	0b1010
Data 1 Echo	Bidirectional	Send SDA bit value 1 received correctly (sent at least 3 times).	0b1011
Data Received Echo	Bidirectional	Indicates that Data Received event was correctly received. It is used to differentiate between consecutive Data 0/1 events during byte transmission (sent at least 3 times).	0b1100
Reserved	Reserved	Reserved for future use.	0b1101-0b1111

Each I2C bus is equivalent to 4-bit payload data as shown in Table 4.3. In the example in Table 4.2, for six I2C buses, three bytes of payload are allocated for I2C channel per frame correspondingly.

Table 4.4. I2C Event Payload Mapping

I2C Payload Byte	Bit Field							
	7	6	5	4	3	2	1	0
0	I2C Bus 1				I2C Bus 0			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
1	I2C Bus 3				I2C Bus 2			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
2	I2C Bus 5				I2C Bus 4			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]

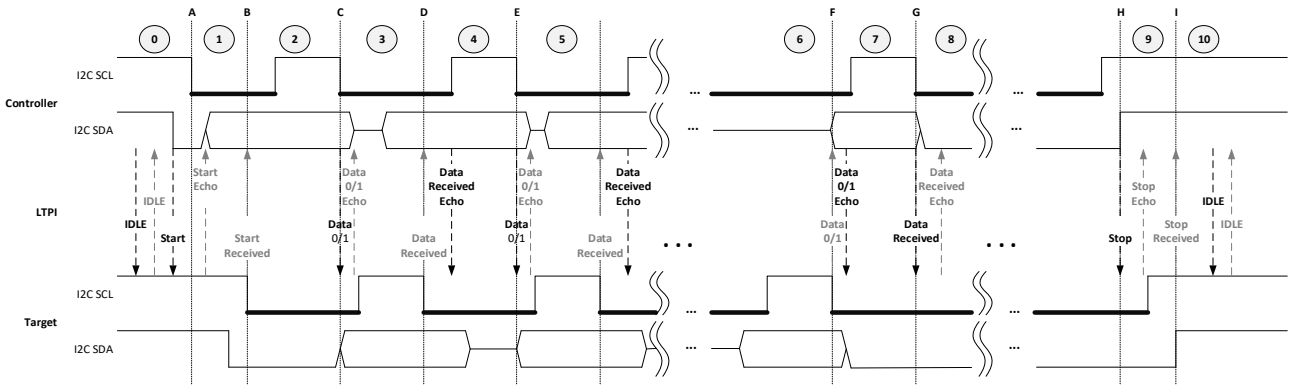


Figure 4.5. I2C Bus Event Exchange Between Controller and Target

Figure 4.5 shows the I2C bus event exchange. The echo support is required by the IP. The event is initiated by the Controller. Once *event request* is detected in LTPI, the communicating party acknowledges receipt by transmitting back the encoded *event echo* to the requestor through LTPI, and when event is successful, it transmits *event received* packets. While waits for feedback from the communicating party, clock stretching is done on local SCL bus to prevent new transactions while the current one is on-going. Encoded packets are continuously transmitted until new event is generated. Bold lines in SCL interface indicate the Clock Stretching done on the SCL bus.

Table 4.5. I2C Event Request Flow

Phase	I2C IP		LTPI	
	Controller	Target	Controller Event	Target Event
0	IP detects SDA falling edge while SCL is high on local bus (*START condition on local bus*). FSM sends START Event to LTPI.	Idle State	IDLE	IDLE
A	• IP detects SCL falling edge on local bus. • IP triggers SCL stretching on local bus and continuously sends START event while waiting for START Received.	Idle State	START	IDLE
1	Clock Stretching	• FSM detects START Event on LTPI. • FSM sends START Echo to LTPI to acknowledge receipt. • FSM generates a START condition signal on SCL and SDA local bus while continuously sending START Echo on LTPI.	START	START Echo

Phase	I2C IP		LTPI	
	Controller	Target	Controller Event	Target Event
B	Clock Stretching	• SCL falling edge is correctly generated on local bus. • FSM sends START Received to LTPI. • IP triggers SCL stretching on local bus and continuously sends START Received while waiting for new event.	START	START Received
2	• FSM detects START Received on LTPI and stops clock stretching on local bus. • Controller starts sending the data bit by triggering raising edge on the bus. • IP detects SCL rising edge and SDA data bit on local bus.	Clock Stretching	START	START Received
C	• IP detects SCL falling edge and FSM sends a Data0/1 Event on LTPI. • IP triggers SCL stretching on local bus and continuously sends Data0/1 Event while waiting for Data0/1 Echo.	Clock Stretching	Data0/1	START Received
3	Clock Stretching	• FSM detects Data0/1 Event on LTPI. • FSM sends Data0/1 Echo to LTPI to acknowledge receipt. • FSM generates a Data0/1 Event on SDA local bus and releases SCL from clock stretching condition while continuously sending Data0/1 Echo on LTPI.	Data0/1	Data0/1 Echo
D	Clock Stretching	• SCL falling edge is correctly generated on local bus. • FSM sends Data0/1 Received to LTPI. • IP triggers SCL stretching on local bus and continuously sends Data0/1 Received while waiting for new event.	Data0/1	Data0/1 Received
4	• FSM detects Data0/1 Received on LTPI. • FSM sends Data Received Echo to LTPI to acknowledge receipt and stops clock stretching on local bus. • Controller starts sending the data bit by triggering raising edge on the bus. • IP detects SCL rising edge and SDA data bit on local bus.	Clock Stretching	Data Received Echo	Data0/1 Received

Phase	I2C IP		LTPI	
	Controller	Target	Controller Event	Target Event
E	• IP detects SCL falling edge and FSM sends a Data0/1 Event on LTPI. • IP triggers SCL stretching on local bus and continuously sends Data0/1 Event while waiting for Data0/1 Echo.	Clock Stretching	Data0/1	Data0/1 Received
5	Clock Stretching	1. FSM receives a Data Received Echo event on LTPI first; this is an indication that of end of previous bit transmission is done. 2. FSM detects Data0/1 Event on LTPI. 3. FSM sends Data0/1 Echo to LTPI to acknowledge receipt. 4. FSM generates a Data0/1 Event on SDA local bus and releases SCL from clock stretching condition while continuously sending Data0/1 Echo on LTPI.	Data0/1	Data0/1 Echo
…	…	…	…	…
Change in direction of data transaction, such as ACK/NACK in this example, but it is the same for Data Bits in Read Transaction.				
…	…	…	…	…
6	Clock Stretching	• Target starts sending the data bit by triggering raising edge on the bus. • IP detects SCL rising edge and SDA data bit on local bus.	Data Received Echo	Data0/1 Received
F	Clock Stretching	• IP detects SCL falling edge and FSM sends a Data0/1 Event on LTPI. • IP triggers SCL stretching on local bus and continuously sends Data0/1 Event while waiting for Data0/1 Received.	Data Received Echo	Data0/1
7	• FSM detects Data0/1 Event on LTPI. • FSM sends Data0/1 Echo to LTPI to acknowledge receipt. • FSM generates a Data0/1 Event on SDA local bus and releases SCL from clock stretching condition while continuously sending Data0/1 Echo on LTPI.	Clock Stretching	Data0/1 Echo	Data0/1
G	• SCL falling edge is detected on local bus. • FSM sends Data0/1 Received to LTPI. • IP triggers SCL stretching on local bus and continuously sends Data0/1 Received while waiting for new event.	Clock Stretching	Data0/1 Received	Data0/1

Phase	I2C IP		LTPI	
	Controller	Target	Controller Event	Target Event
8	Clock Stretching	• FSM detects Data0/1 Received on LTPI. • FSM sends Data Received Echo to LTPI to acknowledge receipt.	Data0/1 Received	Data Received Echo
…	…	…	…	…
H	• IP detects SDA rising edge while SCL is high on local bus (*STOP condition on local bus*). • IP sends STOP event to LTPI. • STOP Event is continuously sent to LTPI while waiting for STOP Received.	—	STOP	Data0/1 Received/ Data Received Echo
9	Last local bus state (IDLE on local bus). • FSM Waits for Stop Echo and Stop Received Event. • IP waits for new START condition. If new START condition is detected, it should be deferred (using clock stretching) until Stop Received event is received for previous STOP condition.	• FSM detects STOP Event on LTPI. • FSM sends STOP Echo to LTPI to acknowledge receipt. • FSM generates a STOP condition signal on SCL and SDA local bus while continuously sending STOP Echo on LTPI.	STOP	STOP Echo
I	Last local bus state (IDLE on local bus). Signal Detection Module waits for new START condition. If new START condition is detected, it should be deferred until Stop Received event is received for previous STOP condition.	• SDA falling edge with SCL high is correctly generated on local bus. • FSM sends STOP Received to LTPI. • IP continuously sends STOP Received.	STOP	STOP Received
10	FSM detects STOP Received on LTPI.	Last local bus state (IDLE on local bus)	IDLE	IDLE

In any case that the LTPI controller or target detects invalid sequence of events, the I2C Code Error interrupt status asserts to indicate error status. When this flag is asserted, I2C bus may encounter hang-up or timeout depending on when the error is detected. The bus can be recovered by triggering I2C channel reset and/or using the built-in timer. If the built-in timer is enabled, the IP performs automatic recovery when error is encountered. If the built-in timer is not enabled, you must perform I2C channel reset to recover the bus during hang-up. Refer to the Bus Timeout and Recovery section for more details.

4.2.1. Glitch Filter

IP Core has integrated glitch filter to remove 50 ns noise/spike as recommended by the I2C Bus Spec for Standard and Fast modes. The glitch filter is applied to both the SCL and SDA signals before they are fed to internal logic. Thus, the I2C signals seen by the IP Core is delayed by a number of clock cycles (~50 ns +1 clock cycle). The filter depth is automatically adjusted based on the input system clock. Due to implementation of glitch filter and nature of I2C tunneling logic, minimum of 10 MHz input system clock is required for the IP with I2C 100 kHz channel enabled and 20 MHz input system clock for I2C 400 kHz channel enabled to work correctly (for example, 50 MHz DDR or 100 MHz SDR LVDS clock and above for I2C 100 kHz mode, 100 MHz DDR or 200 MHz SDR LVDs clock and above for I2C 400 kHz mode). The IP with I2C channel enabled is not guaranteed to work if input system clock frequency is less than the minimum required for each mode. For list of I2C support limitation, refer to Appendix B.

4.2.2. Bus Timeout and Recovery

For special cases wherein a defective device is holding the clock low indefinitely or the bus has already stalled, the IP supports 1) a built-in timer that monitors the bus and automatically performs recovery sequence once timeout is detected and 2) a manual recovery through a separate I2C channel reset (Link Control [8:2] register).

4.2.2.1. Built-in Timer

When *Enable Timer[I2C index]* == 1, a built-in timer is automatically added for the corresponding bus. It monitors SCL and when observed that it is held low for *Timeout Value*, it automatically starts recovery sequence.

Every time SCL goes low, timer starts and resets once it detects a rising edge of the clock. When SCL is held low for *Timeout Value* , IP asserts I2C Timeout detected status and starts recovery.

I2C Target Interface

For *I2C Target*, the IP immediately releases SCL and SDA and resets the internal controller, that is, the IP stops driving SCL and SDA and let them float high. After recovery, LTPI waits for START to begin processing transactions again. All transactions prior to START are ignored.

I2C Controller interface

For *I2C Controller*, the IP attempts generating a STOP event. It waits for the correct STOP event to be observed on the bus. Figure 4.6 shows the sample timing diagram when the bus is successfully recovered in the first attempt.

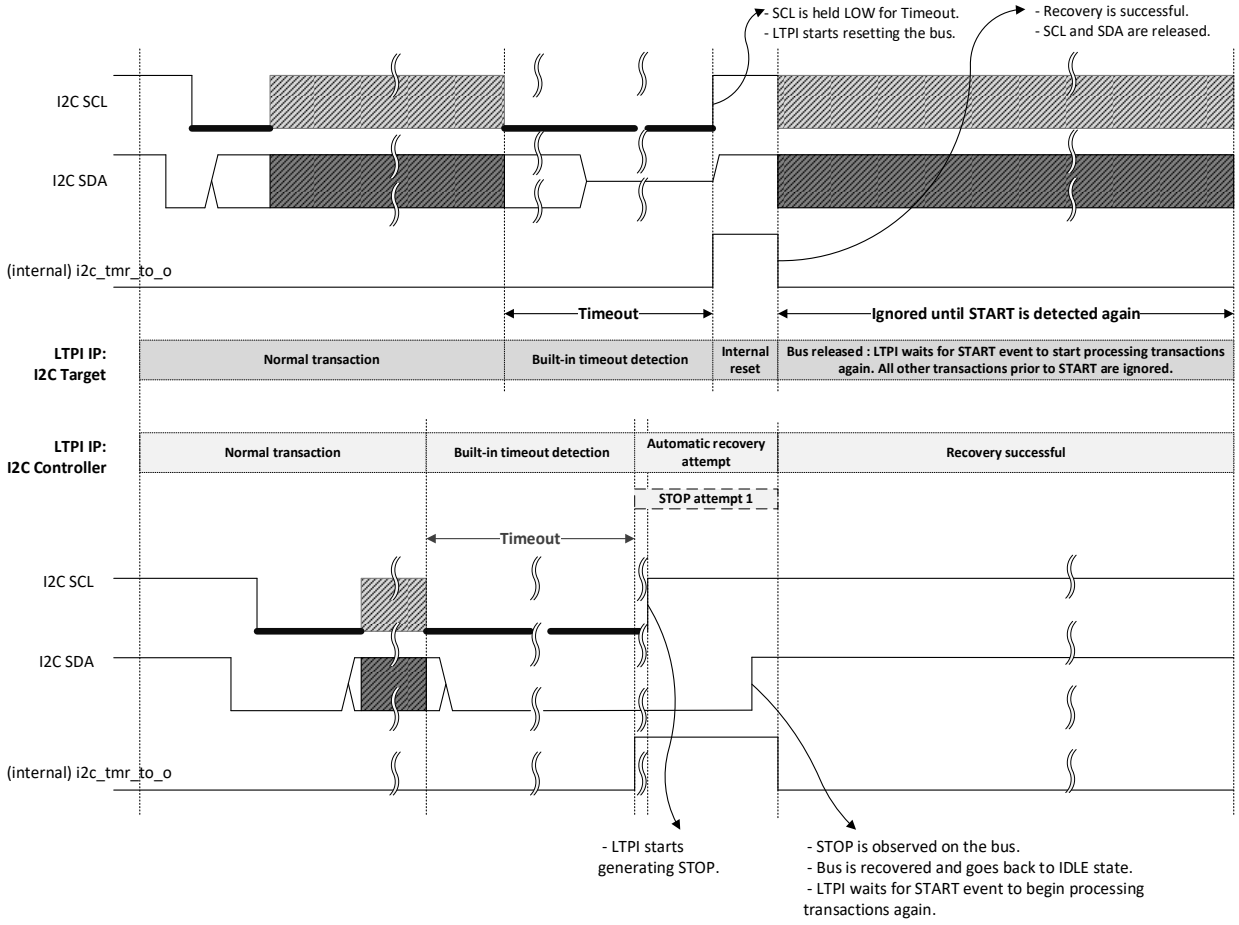


Figure 4.6. Bus Recovery with Built-in Timer and Successful STOP Generation in the First Attempt

If after *Timeout* and the bus is still not as expected, that is, STOP is not observed on the bus, it holds SCL low for at least another *Timeout* period and then generates another STOP event to reset the bus interface again. The cycle continues until the correct event is observed on the bus. If STOP is observed on the bus, recovery is successful. Refer to Figure 4.7 for illustration.

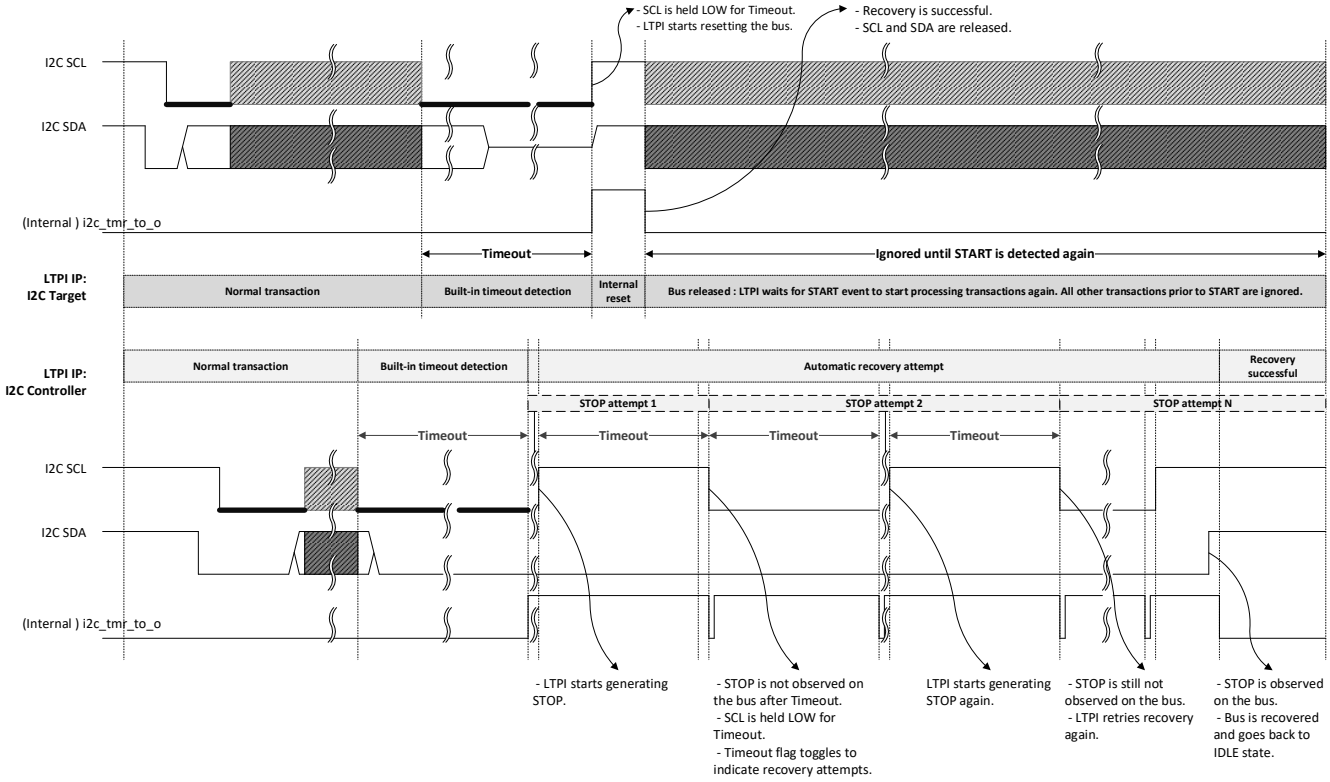


Figure 4.7. Bus Recovery with Built-in Timer and Automatic Retry Attempts

4.2.2.2. I2C Channel Reset

The IP provides a separate I2C channel reset through the Link Control [8:2] register to manually reset the internal I2C controller of the IP to recover the bus in case of timeout or bus hang-up. This register is always available, but the reset behavior differs depending on *I2C Bus interface* and availability of the built-in timer in the IP.

I2C Target

For *I2C Target*, during the assertion of I2C channel reset, the IP immediately releases SCL and SDA and resets the internal controller. This means the IP stops driving SCL and SDA, allowing them to float high.

I2C Controller

For *I2C Controller,* falling edge detection (release) of the I2C channel reset triggers STOP event generation on the bus and resets the internal I2C controller of the IP. During STOP generation, SCL is held low for at least the minimum SCL low period (T_{LOW}). SCL is held high after this period to generate the STOP condition.

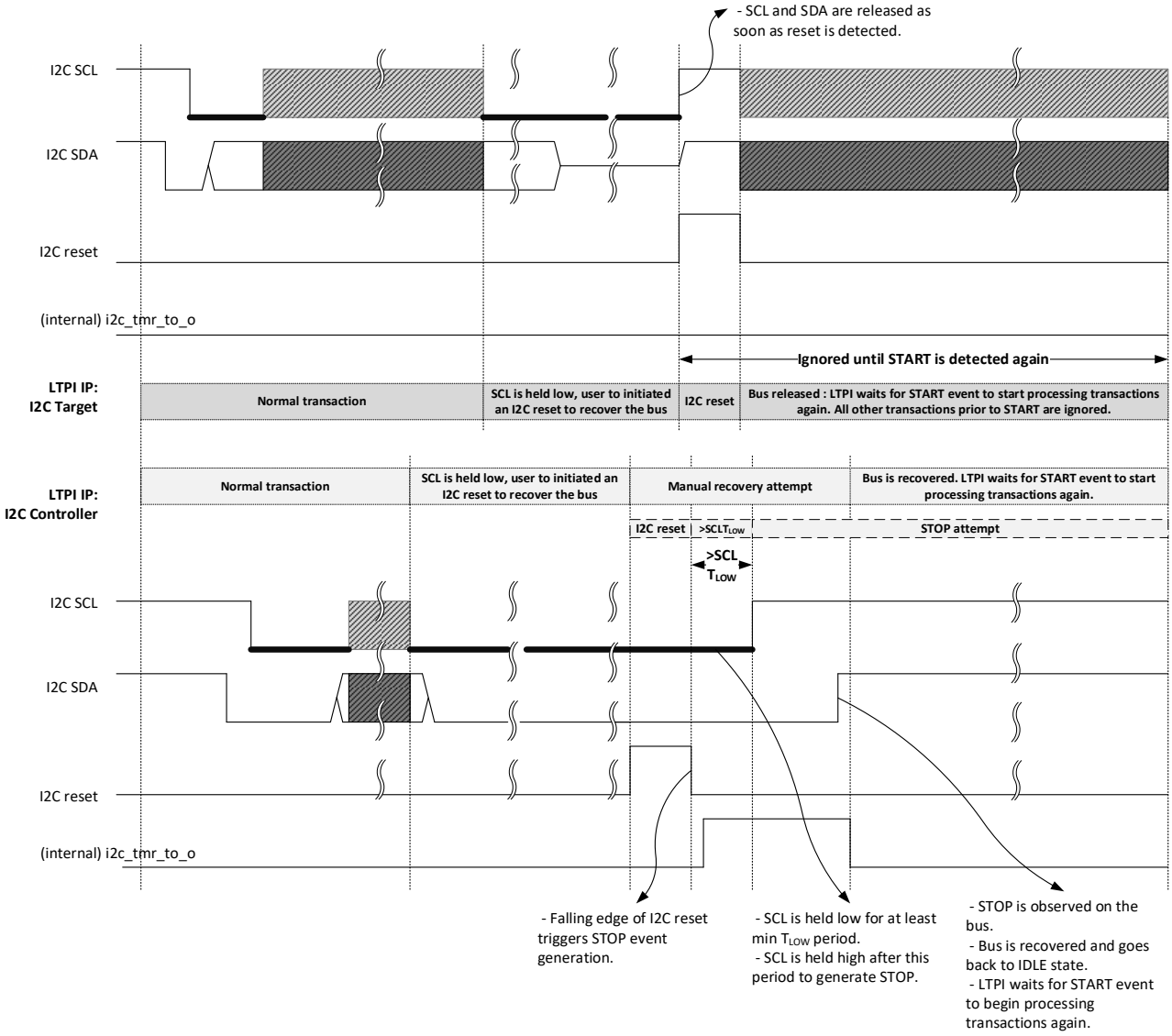


Figure 4.8. I2C Channel Reset with Manual Reset for LTPI Interfacing with Target Device

In the case that STOP is not properly observed on the bus and the status register I2C Timeout detected is still high after the initial reset, you can manually trigger another I2C channel reset by toggling the register again. Each falling edge detection of I2C channel reset triggers STOP event generation on the bus and resets the internal I2C controller of the IP.

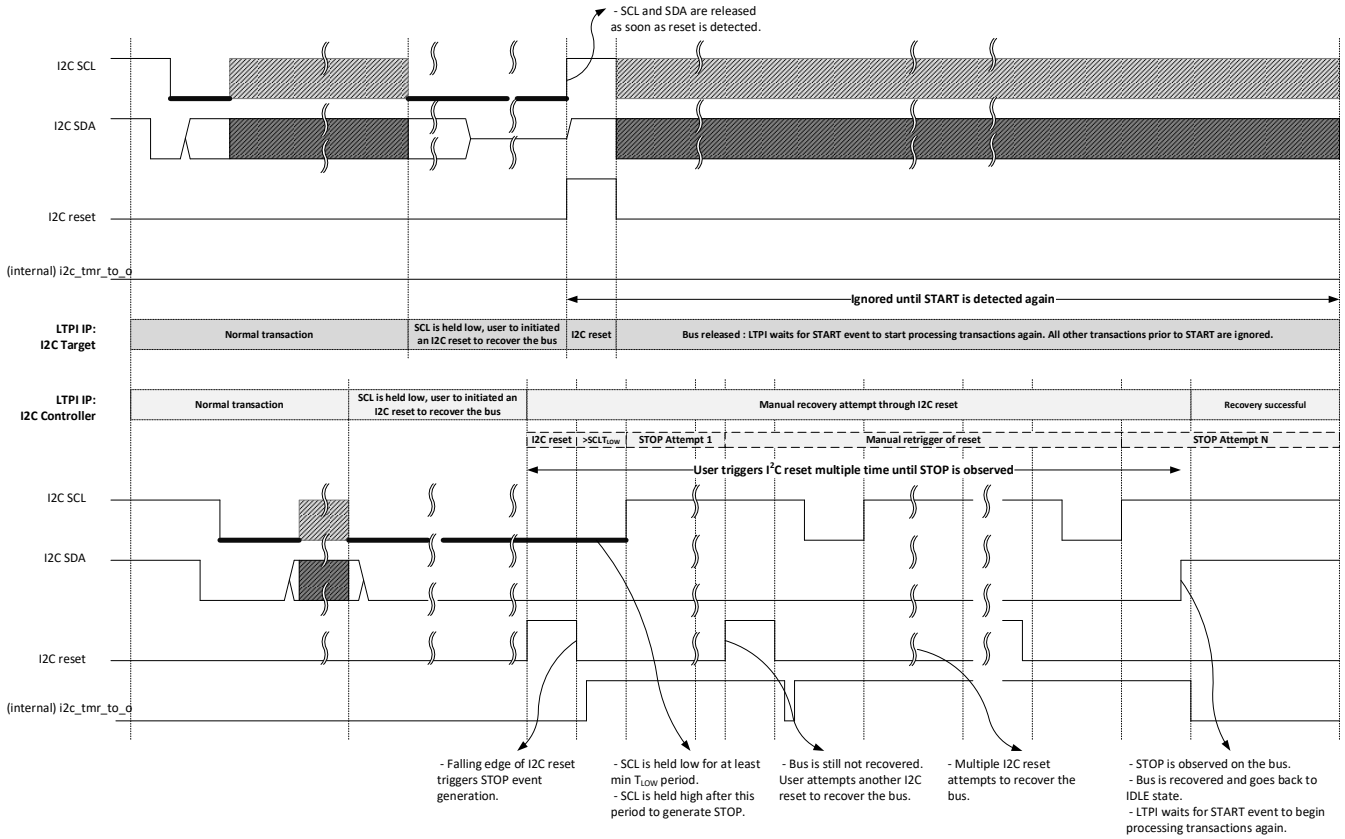


Figure 4.9. Channel Reset with Multiple Manual Reset Attempts for LTPI Interfacing with Target Device

For a special case that an external target device holds SDA low even after I2C channel reset sequence, it is recommended to use the built-in timer to be able to perform continuous recovery retries.

Table 4.6. Summary of Bus Timeout and Recovery Feature

LTPI	I2C Reset		Built-in Timer	
	Release on Reset	Generate Stop on Reset	Release on Reset	Generate Stop on Reset
LTPI IP Interfacing with External Controller Device (*I2C Target*)				
Timer Enabled (*Enable Timer[I2C index]* == 1)	Yes	N/A	Yes	N/A
Timer Disabled (*Enable Timer[I2C index]* == 0)	Yes	N/A	N/A	N/A
LTPI IP Interfacing with External Target Device (*I2C Controller*)				
Timer Enabled (*Enable Timer[I2C index]* == 1)	N/A	Yes, with manual retry	No	Yes, with automatic retry
Timer Disabled, (*Enable Timer[I2C index]* == 0)	N/A	Yes, with manual retry	N/A	N/A

4.2.3. Bidirectional I2C

The Management Component Transport Protocol (MCTP) can be supported over I2C channels. The MCTP function requires a bidirectional way of transfer and thus requires that I2C channels will have both a Controller function and a Target function that can switch dynamically. That is, the I2C channels must have an arbitration capability just like in multi-controller topology.

The current DC-SCM LTPI standard does not support tunneling of I2C transfer with arbitration (multi-controller). To support this feature, the I2C event encoding and protocol is changed during the address phase after the I2C START to allow the arbitration process. Figure 4.10 shows the timing diagram and exchange of LTPI I/O frames during I2C Start and Address Phase.

The external I2C device connected on both sides of the LTPI may initiate an I2C transaction and it is also possible to happen at almost the same time. Initially (right after I2C Start), all I2C channels are active to receive and transmit. The I2C arbitration process is done after I2C Start and the direction of the transfer is determined based on which device wins the arbitration (either I2C Device 0 or I2C Device 1).

After the arbitration, only one I2C channel is active and proceeds with the transfer just like a normal I2C transaction. Note that the arbitration is only performed during the address phase after the I2C Start symbol. MCTP devices must have a unique I2C address; otherwise, the direction of transfer may not be resolved and may produce an undesirable result.

When both I2C devices simultaneously initiate a transaction, the LTPI I2C event FSM transmits and receives a Start event. In this case, since both I2C interfaces have already started, they can proceed to send the first address bit (MSB) after waiting for at least half an SCL period. This ensures that the value of the data bit is already stable. After sending and receiving Data 0/1 event, the data value is driven to both interfaces.

The SDA line that is pulled down does not change but if it is high, it may get pulled down or remain high depending on the received Data 0/1 event. Then both sides send Data 0/1 echo to confirm the result which is expected to have the same value. The SCL line is now released to proceed to the next event. The Data Received event is sent to indicate the end of the SCL cycle (after falling edge). Unlike in the original protocol, the Data Received Echo event is not necessary as both LTPI receive the Data Received event.

The next bit starts after waiting for at least half an SCL cycle and sending Data 0/1 event for the next address bit. This will continue until the eighth bit, which is the Read/Write bit. Both external I2C devices know which device wins the arbitration – the first device that drives a high will loses the arbitration, while the device that drives low wins and assumes the role of I2C Controller. Before proceeding to the ACK/NAK phase, the LTPI I2C event FSM evaluates and resolves the arbitration.

The last event after the Read/Write bit and before ACK/NAK phase is sending and receiving Data Received event. The side that wins the arbitration is the one that sends the Data Received Echo event. At this point, the original protocol is followed and the external I2C devices temporarily assume their role (as controller or target depending on which device wins the arbitration) until the transaction is terminated.

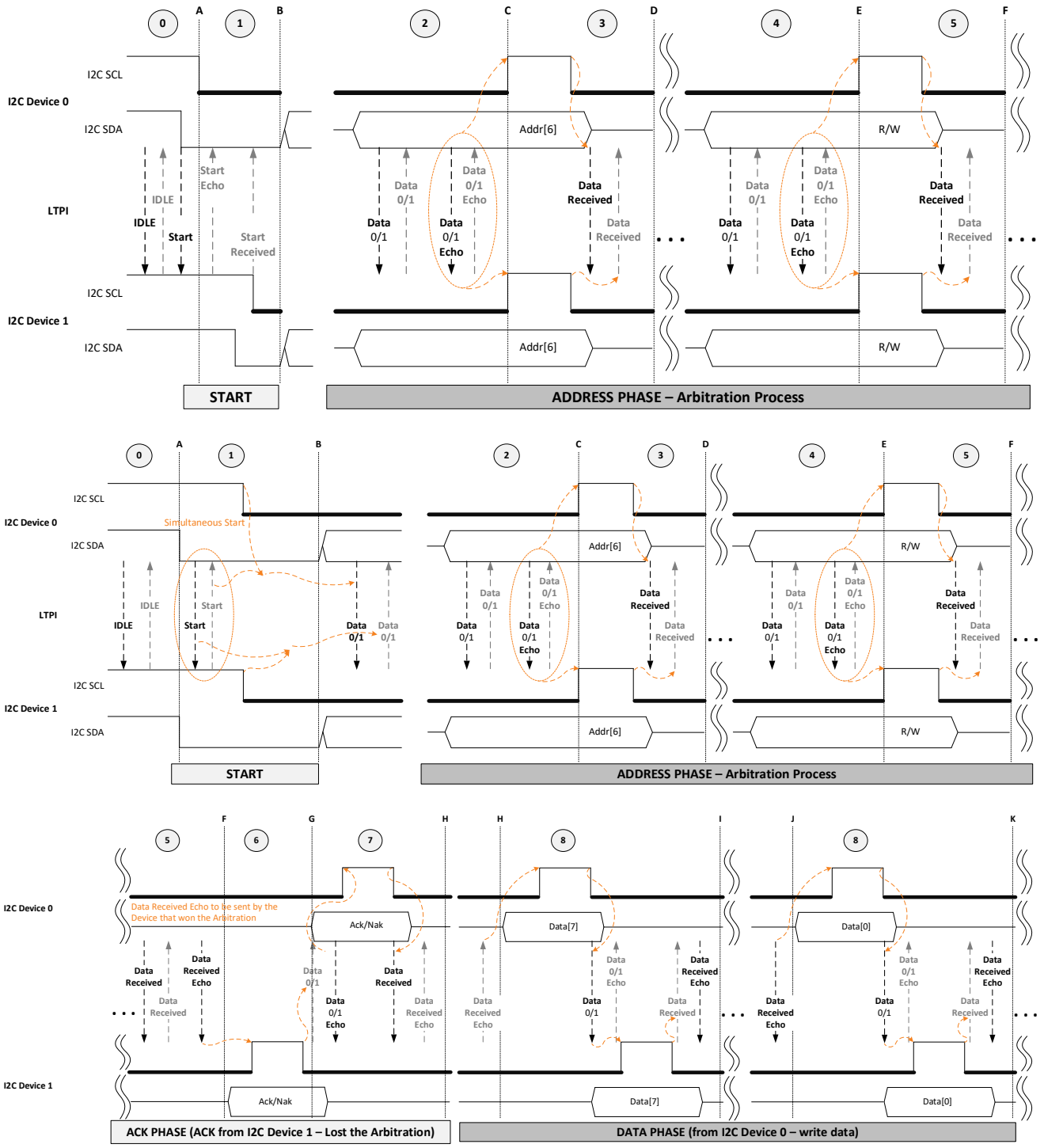


Figure 4.10. I2C Start and Arbitration Process in Bidirectional I2C

4.3. UART Channel

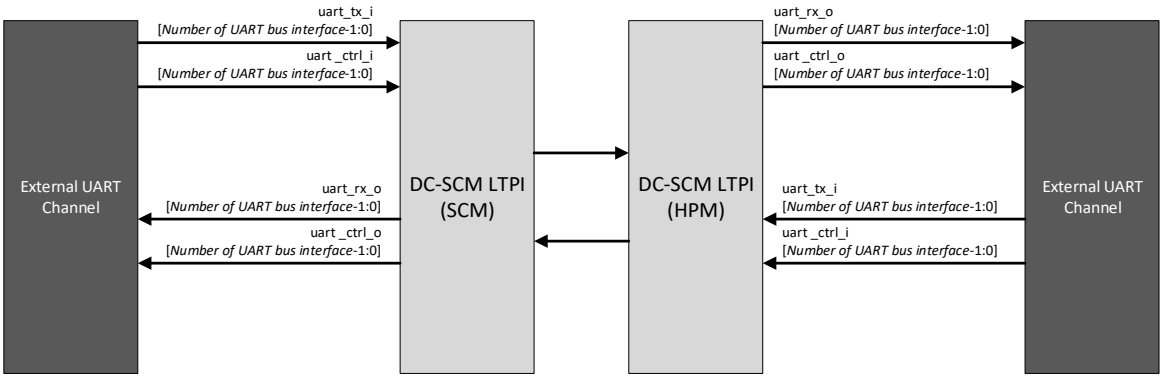


Figure 4.11. UART Pin List

The UART channel is used to tunnel physical UART interfaces between SCM and HPM through LTPI. The UART channel supports tunneling of multiple full-duplex UART interfaces with flow control signals.

UART mechanism is similar to LL GPIO except that UART Transmit Data and Receive Data lines are oversampled by a factor of three and the three consecutive samples are being encoded in every LTPI Frame as shown in Figure 4.12.

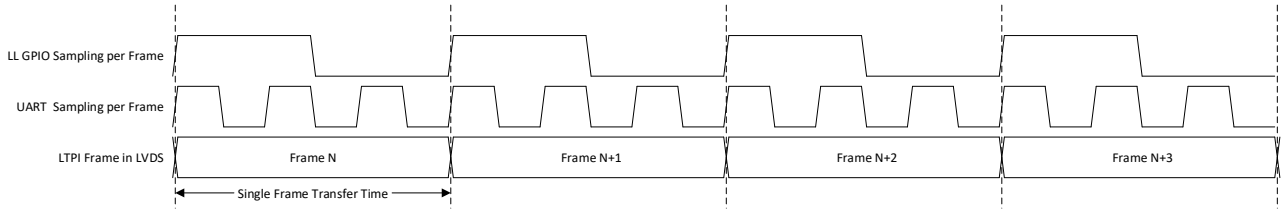


Figure 4.12. UART General Oversampling Principle

To maintain a sampling duty cycle close to 50%, UART sampling is done every four system clock cycle intervals in every frame as shown in Figure 4.13. The same sampling distribution is implemented in regenerating the UART data.

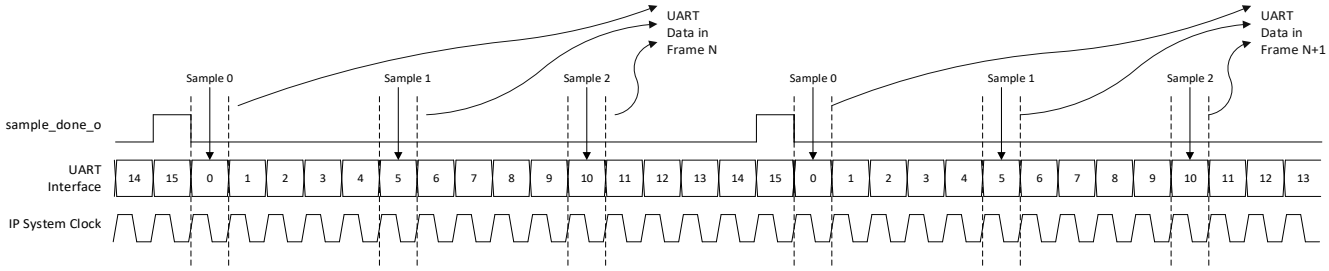


Figure 4.13. LTPI UART Sampling Distribution

Each UART bus is allocated 4-bit in the frame. Payload allocation for both Rx and Tx frames depends on the *UART Byte Size* attribute set.

Table 4.2 shows one byte is allocated for UART, which is equivalent to two UART buses. Table 4.7 lists the content of each UART bus.

Table 4.7. UART Bus Content

Bus Number	Bit Position	Description
UART Bus 0	0	uart_tx_i[0] / uart_rx_o[0] – Sample 0 (UART TXD to RXD bus line)
	1	uart_tx_i[0] / uart_rx_o[0] – Sample 1 (UART TXD to RXD bus line)
	2	uart_tx_i[0] / uart_rx_o[0] – Sample 2 (UART TXD to RXD bus line)
	3	uart _ctrl_i[0] / uart _ctrl_o[0] – UART CTS/RTS line
UART Bus 1	4	uart_tx_i[1] / uart_rx_o[1] – Sample 0 (UART TXD to RXD bus line)
	5	uart_tx_i[1] / uart_rx_o[1] – Sample 1 (UART TXD to RXD bus line)
	6	uart_tx_i[1] / uart_rx_o[1] – Sample 2 (UART TXD to RXD bus line)
	7	uart _ctrl_i[1] / uart _ctrl_o[1] – UART CTS/RTS line

For the Tx, IP samples the UART input ports. If there is a total of two UART buses and *UART Byte size* is set to 1 byte, IP samples the two buses.

For the received frames, IP remaps the received payload into the UART output pins, that is, uart_tx_i[0] to uart_rx_o[0], uart _ctrl_i[0] to uart _ctrl_o[0], and so on. The IP follows the same Tx multiplexing mechanism. The IP distributes data to each of the two UART buses.

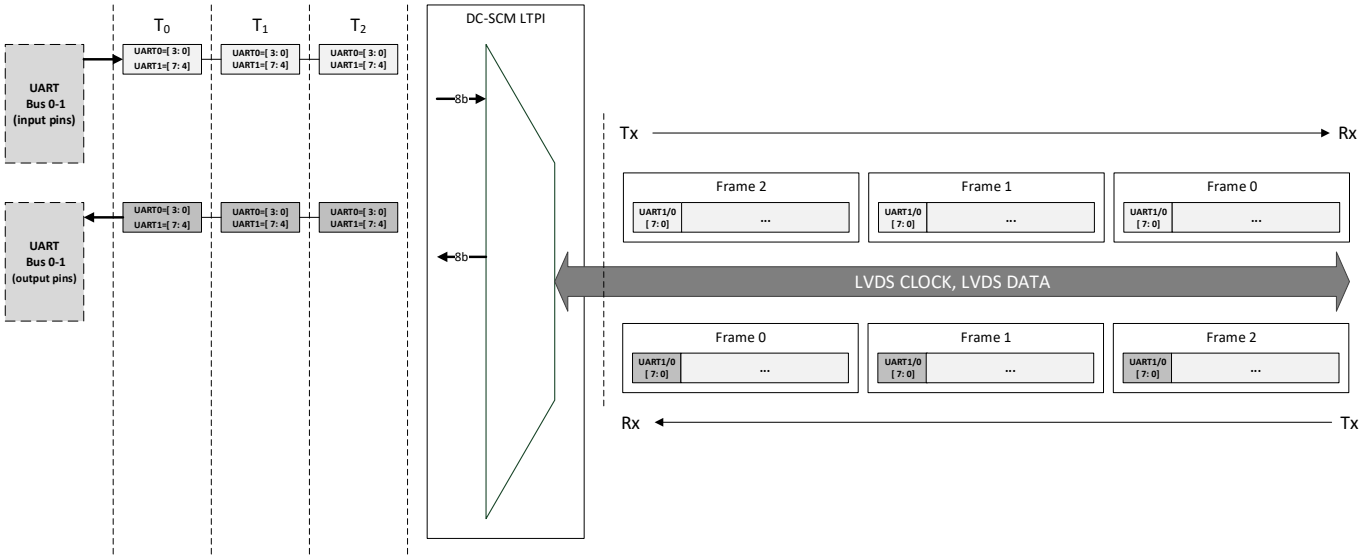


Figure 4.14. UART Mechanism

4.4. OEM Channel

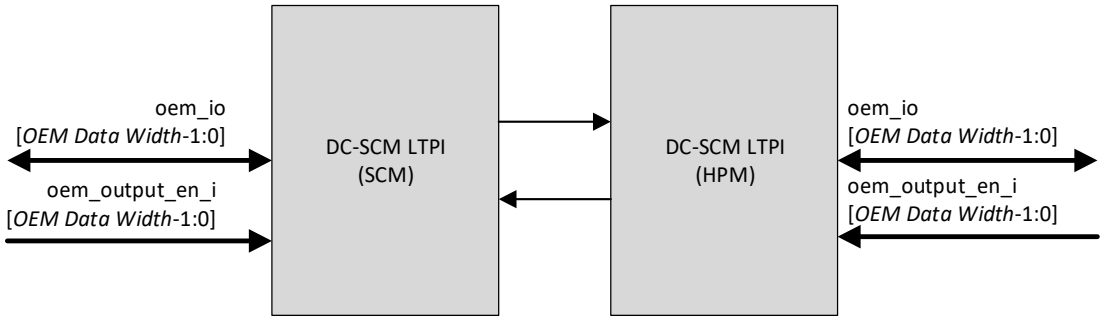


Figure 4.15. OEM Pin List

For OEM, pins are set as bidirectional and payload allocation for both Rx and Tx frames depends on the *OEM Byte Size* attribute set. It follows sampling mechanism similar to LL GPIO except that when enabled, OEM pins are automatically mapped to IO buffers and interface is updated only every non-Data Channel frame transfer. The IP samples the OEM ports based on *OEM Byte Size* attribute set. Table 4.2 shows the two bytes are allocated for OEM. If there is a total of 16 *OEM Data Width*, IP samples the 16-bit OEM pins per sampling time.

Regardless of the direction, Tx always samples the OEM bidirectional pins when creating the Tx packet. Similarly, Rx always sends out the received OEM data to OEM bidirectional pins. It is up to you to control the intended direction of the I/O through oem_oe_i. When this port is high, I/O is set as an output. When this port is low, I/O is set as input.

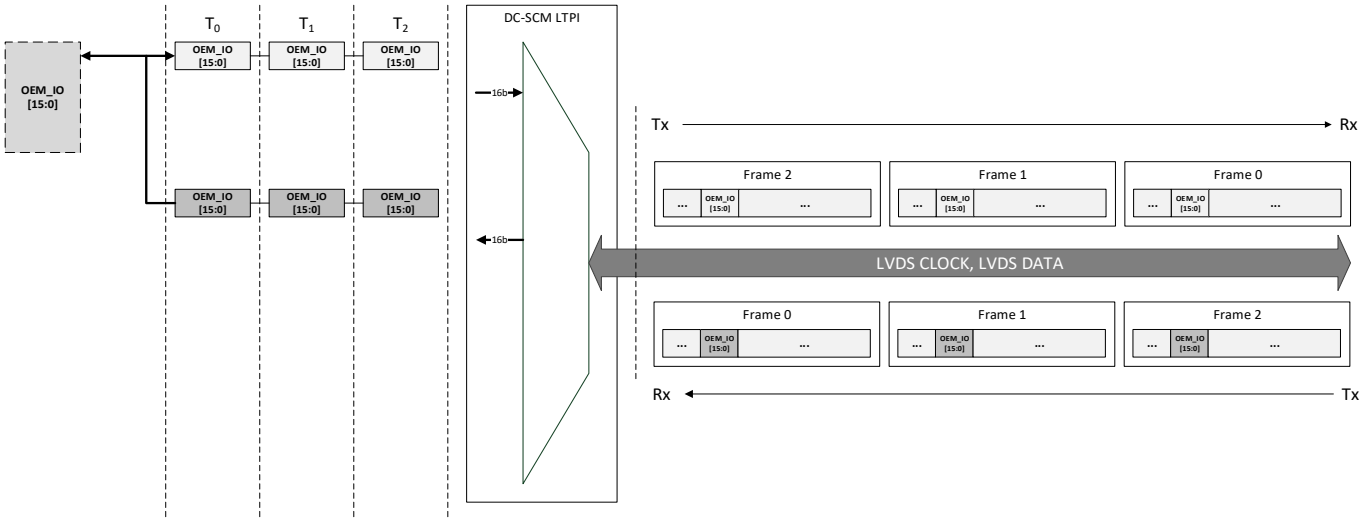


Figure 4.16. OEM Mechanism

4.5. Data Channel

Data Channel allows tunneling of standardized Read and Write requests to addressable memory spaces. The DC-SCM LTPI soft IP supports the mapping of Data Channel to APB interface for SCM mode and can be accessed on-demand when enabled with BMC always acting as the initiator of requests. SCM always acts as the pseudo-completer device, and HPM always acts as pseudo-requester. The typical usage of Data Channel is for the SCM side to access the registers on the HPM side.

Data Channel Interface uses the same APB interface for CSR access and is provided a region within the CSR address space. APB access that falls under the Data Channel space is converted to the appropriate encoded packet and tunneled to the HPM side. HPM provides the necessary response to the Data Channel access.

It is also possible for the APB requester from the HPM side to access the SCM through the Data channel, but this is not supported in this version.

The memory region or space accessible by the SCM on the HPM is indicated in the IP GUI setting *Data Channel Memory Size*. By default, the *Data Channel Memory Size* is 4 KiB, which means only the CSR space of the HPM is accessible. In HPM, if external access through Data Channel is desired, you can enable the *Data Channel: Enable External Access* option in the IP GUI. This adds a separate APB requester interface.

The *Data Channel Memory Size* (in both the HPM and SCM) must be set to the address space region accessible through the APB requester interface, plus the size of the CSR space, and rounded to the nearest power of 2. For example, 8 KiB external space + 4 KiB CSR space = 12 KiB. The nearest power of 2 is 16 KiB, thus *Data Channel Memory Size == 16*.

From the SCM-side APB, the SCM CSR is accessible at base offset 0x0000, HPM CSR at base offset 0x1000, and the external space at base offset 0x2000 and beyond.

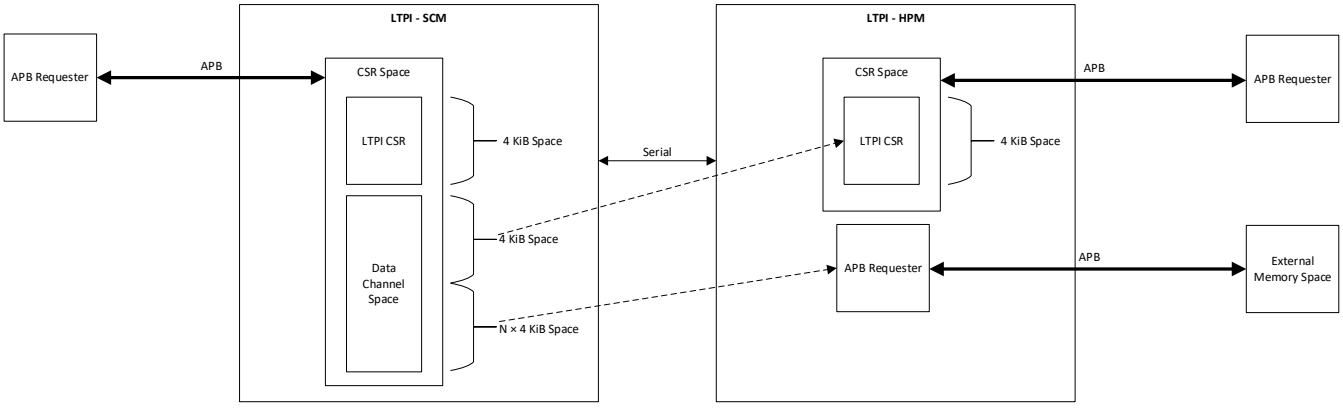


Figure 4.17. APB Interface for SCM and HPM

Table 4.8. Data Channel Command Encoding

Command	Parameters	Description	8-bit Event Encoding
Read Request	Address	Command indicates a request to read data from specific address, such as a register address.	0x00
Write Request	Address and Data	Command indicates a request to write data to a specific address, such as a register address.	0x01
Read Completion	Address and Data	Command indicates a read operation is completed.	0x02
Write Completion	Address	Command indicates a write operation result.	0x03
CRC Error	N/A	Command indicates a Data Frame was received with CRC error and the operation would not be completed.	0x04
Reserved	Reserved	Reserved	0x05–0xFF

Depending on the command request, each data channel byte is mapped to each payload byte in the data frame.

Table 4.9. Memory Command Frame Mapping

Payload Byte Sequence	Map	Corresponding Value	Description
0	data_ch[7: 0]	8'h00	Tag field that can be used for customer specific purposes such as tracking multiple outstanding requests. This is not supported.
1	data_ch[15: 8]	Command	8-bit Command Encoding. For more details, refer to Table 4.8.
2	data_ch[23:16]	Address Byte 3	Address for the request as indicated in the APB interface (SCM). For CRC Error command frame, this takes last successful known address. For requests that hits the HPM register space, the SCM APB base address is subtracted and only the offset (11:0) is forwarded to HPM. For example, access to HPM Link Status register at 0x1200, would have this address set to 0x200.
3	data_ch[31:24]	Address Byte 2	
4	data_ch[39:32]	Address Byte 1	
5	data_ch[47:40]	Address Byte 0	
6	data_ch[55:48]	{Operation Status ,Byte Enable}	Refer to Table 4.10.
7	data_ch[63:56]	Data Byte 3	Data bytes for Write Request and Read Completion commands as indicated in the APB interface (SCM). For Read Request, Memory Write Completion, and CRC Error commands, these data bytes are set to 0.
8	data_ch[71:64]	Data Byte 2	
9	data_ch[79:72]	Data Byte 1	
10	data_ch[87:80]	Data Byte 2	

Table 4.10. Data Mapping for Operation Status and Byte Enable of Data Channel

Operation	Bit Field for data_ch[55:48]							
	7	6	5	4	3	2	1	0
Memory Read	Reserved				Byte Enable[1], fixed to 0xF			
Memory Read Completion	Read Operation Status[2] 0x0: Success 0x1: Invalid Access 0x2–0xF: Reserved				Byte Enable[1], fixed to 0xF			
Memory Write	Reserved				Byte Enable[1], fixed to 0xF			
Memory Write Completion	Write Operation Status[2] 0x0: Success 0x1: Invalid Access 0x2–0xF: Reserved				Reserved			
CRC Error	Reserved							

Notes:

1. All 4-byte data are considered valid.
2. For read and write operation completion status, status is processed from PSLVERR input to APB Requester interface.

The time it takes to finish a complete end-to-end APB transaction varies depending on the system clock frequency used and target device response. The APB completer interface in SCM extends the transfer by driving APB_READY to LOW until it receives the response from HPM. This means that only one Data Channel transaction is processed by SCM at a time and that transaction should either be completed or terminated in order to process a new request.

4.5.1. CRC Error Command Handling

When CRC error command is received, operation on data bus cannot be executed. The LTPI side which detected the CRC error should drop the frame and inform the remote side of the CRC error received by sending the dedicated Data Channel Command indicating CRC error. The other LTPI side can decide on retransmission or return error to the originator of Data Channel access. Table 4.11 lists the summary of how IP processes CRC error for Data Channel.

Table 4.11. Summary of Data Channel CRC Handling

Scenario	SCM	HPM
CRC error/mismatch is detected in SCM	• SCM detects CRC error/mismatch in received Data frame, that is Memory Write/Read Completion command. • SCM drops the frame and generates Data Channel command indicating CRC error, that is 0x04 command. • SCM terminates the current transaction in APB bus and asserts PSLVERR to indicate error in Data Channel access.	• HPM receives Data Channel command indicating CRC error, that is 0x04 command. • HPM asserts Frame CRC error status.
CRC error/mismatch is detected in HPM	• SCM receives Data Channel command indicating CRC error, that is 0x04 command. • SCM terminates the current transaction in APB bus and asserts PSLVERR to indicate error in Data Channel access.	• HPM detects CRC error/mismatch in received Data frame, that is Memory Write/Read request command. • HPM drops the frame and generates Data Channel command indicating CRC error, that is 0x04 command.

5. Clocks and Reset

Below is the list of clocks used by the IP. Figure 5.1 shows the general clock topology.

- Edge clock (eclk_i) and 90-degree phase shifted edge clock (eclk90_i) are clocks used by the Tx DDR soft IPs for gear-related processing. These clocks must be source synchronous and frequency is equivalent to the target LVDS PHY frequency. All CDC transfers in the DDR domain are already handled by the hard IPs instantiated to implement the DDR clock tree.
- SCLK Domain – Both DDR Tx and Rx blocks generate core clock (sclk tx and sclk rx) which are used to clock the read operation of the internal Tx FIFO and write operation of the internal Rx FIFO respectively. Frequency of SCLK is dependent on LVDS PHY mode and DDR gearing used. All CDC transfers in the DDR domain are already handled by the hard Ips instantiated to implement the DDR clock tree.
- System CLK Domain – A system clock (clk_i) is used to clock the write operation of the Tx FIFO and read operation of the Rx FIFO. This clock is also used to clock the transmit and receive related logic of the upper layer, interface channel controllers, and APB.

The edge clock (eclk_i/eclk90_i) requires correct clock switching and must be done externally which includes but is not limited to reconfiguration of the GPLL/clock source. Refer to the Clock Switching section for more details.

The system clock (clk_i) can have a fixed frequency and does not need to follow the gearing ratio of the Edge clock and SCLK as long as it is not below the gearing ratio.

Moreover, in terms of tunneling, it is assumed that external channels operate at much lower frequencies compared to the operating system clock of the IP.

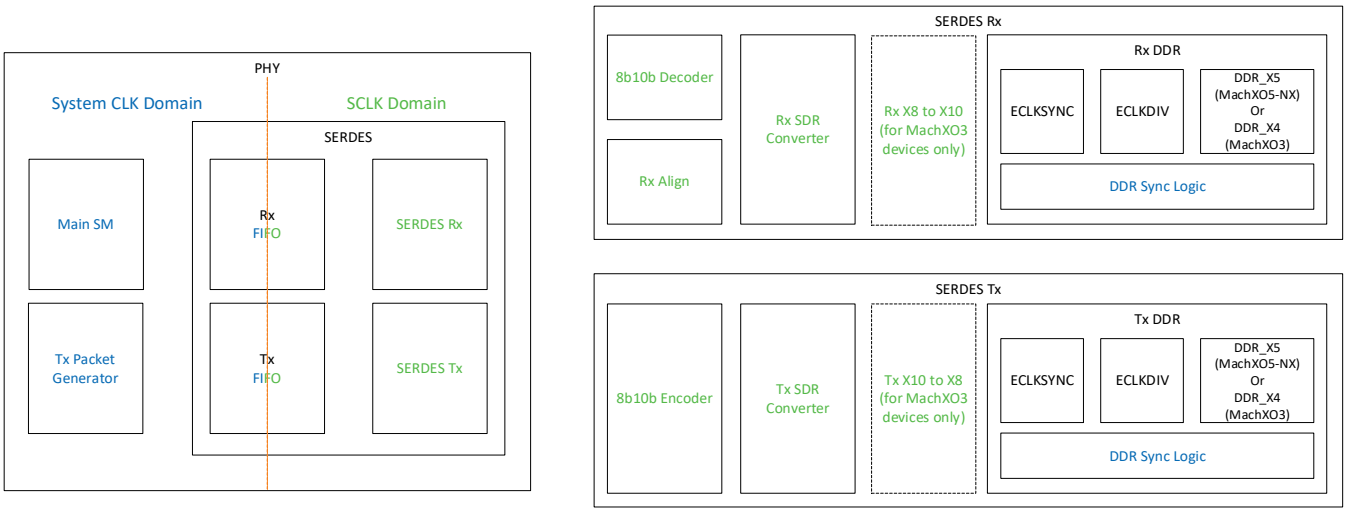


Figure 5.1. General Clocking Topology (Main)

To avoid Rx FIFO overflow or Tx FIFO underflow, it is preferable for the system clock frequency to be slightly higher than the SCLK frequency. To further ensure FIFO stability, the recommended end-to-end clocking implementation between two devices is shown in Figure 5.2.

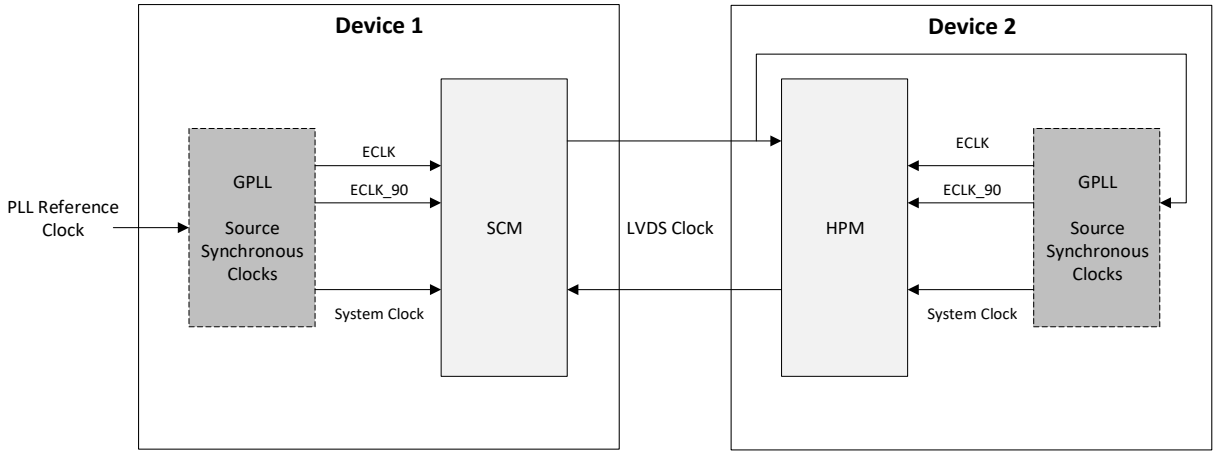


Figure 5.2. End-to-End Clocking Implementation

In the recommended clocking implementation, either SCM or HPM should be set as the Main source of clock. GPLL residing in Device 1 has a unique and independent reference clock as illustrated in the figure. The other device is then set as Follower. The Follower uses the incoming LVDS Rx clock from the Main as the reference clock of its clock generation logic, for example, as reference clock of the PLL residing in the Follower device, as shown in Device 2. Implementing this ensures that clocks are properly synchronized between the two devices, SCM and HPM.

It is up to the system level design to set which is the main source of clock. In cases when Lattice IP is paired with non-Lattice SCM or HPM device, Lattice IP should always be the clock Follower.

5.1. Clock Computations

$LVDS\ Mode = 2\ (DDR); 1\ (SDR)\ bit/s\ per\ clock$

$DDR\ Gear\ = 8\ (MachXO3\ and\ Mach-NX);\ 10\ (MachXO5-NX\ )$

$LVDS\ Clock = PHY\ Data\ Rate\ /\ LVDS\ Mode$

$Edge\ Clock\ = LVDS\ Clock$

$SCLK = LVDS\ Clock\ /\ (DDR\ Gear\ /\ LVDS\ Mode)$

$CLK\_I = PHY\ Data\ Rate\ /\ 10$ (minimum frequency)

5.2. Sample Clock Computations

Below is a sample configuration for 25 MHz SDR LVDS Base Clock Frequency:

- PHY Interface Data Rate = 25 Mbps
- LVDS Mode = 1 (SDR) bit/s per LVDS clock
- DDR Gear = 8 (MachXO3)
- PHY internal clocking
  - LVDS Clock = 25 Mbps / 1 = 25 MHz
  - Edge Clock = 25 MHz
  - SCLK = 25 MHz / (8/1) = 3.125 MHz
- System Clock must match the PHY side throughput
  - CLK_I = 25 Mbps / 10 = 2.5 MHz (minimum)

Below is a sample configuration for 400 MHz DDR LVDS Target Clock Frequency Clock:

- PHY Interface Data Rate = 800 Mbps
- LVDS Mode = 2 (DDR) bit/s per LVDS clock
- DDR Gear = 8 (MachXO3)

- PHY internal clocking
  - LVDS Clock = 800 Mbps / 2 = 400 MHz
  - Edge Clock = 400 MHz
  - SCLK = 400 MHz / (8/2) = 100 MHz
- System Clock must match the PHY side throughput
  - CLK_I = 800 Mbps / 10 = 80 MHz (minimum)

For convenience, the table below shows the recommended system clock frequency for the given range of LVDS frequencies.

Table 5.1. Recommended System Clock Frequency

DDR Enabled		SDR Only	
Max LVDS Clock (MHz)	**System Clock (MHz)**	**Max LVDS Clock (MHz)**	**System Clock (MHz)**
100 (and below)	25	200 (and below)	25
200 (and below)	50	400 (and below)	50
400 (and below)	100	600 (and below)	100
600 (and below)	125	—	—

5.3. Reset

The asynchronous active low reset (rst_n_i) is implemented as the system reset of the IP. A reset synchronizer circuit is already implemented inside the IP.

6. Error Impact, Handling, and Recovery

Table 6.1. IP Error Handling

Scenario	Tx	Rx
Failed Word Alignment	Tx assumes that the receiving host can finish link synchronization/word alignment within the set training period.	Error handling is out of scope by the IP. The IP has a status register (*LTPI Link Aligned*) to indicate if word alignment is achieved.
LTPI FSM Protocol Timeout	The IP issues corresponding timeout flag signal to indicate timeout is encountered.	Error handling is out of scope by the IP. The IP issues interrupt status to indicate timeout. Refer to the Register Description section for more details.
CRC Error	N/A	The IP issues interrupt status to indicate failure. Refer to the Register Description section for more details. The received frame is discarded once Rx detects a wrong CRC in the received frame. For each channel, the CRC error impact is different. Refer to Table 6.2 for more details.

Table 6.2. CRC Error Impact on LTPI Channels

Channels	Impact	Recovery
Low Latency GPIO	The state of LL GPIO is maintained from the previous successful frame. The next frame with correct CRC provides updated states of the LL GPIO.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.
Normal Latency GPIO	The state of subset of NL GPIO is maintained based on previous successful frame. The next update happens after (N-1) number of Frames with correct CRC.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.
I2C	The state of I2C is maintained based on previous successful frame. The next frame with correct CRC provides the updated states of the I2C. In I2C perspective, this condition does not yet affect the local bus since clock stretching is implemented. If CRC error is not permanent, the correct I2C event is decoded in the next correct LTPI frame.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.
UART	The state of UART is maintained from the previous successful frame. The next frame with correct CRC provides the updated states of the UART. From the UART interface perspective, this condition causes jitter if the dropped frame contained the UART signal transition. The correct signal transition happens with the next correct LTPI frame.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.
Data Channel	When CRC error is received, the operation on the Data bus cannot be executed. The LTPI side which detects the CRC error should drop the frame and inform the remote side of the CRC error received by sending the dedicated Data Channel Command indicating CRC error. The other LTPI side can decide on retransmission or return error to the originator of Data Channel access. Refer to Table 4.11 for details.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.

Table 6.3. Timeout Impact on LTPI Channels

Channels	Impact	Recovery
Low Latency GPIO	N/A	N/A
Normal Latency GPIO	N/A	N/A
I2C	If LTPI events are not received by the other end, the I2C stays in its previous state.	The IP supports the following recoveries: • A built-in timer that monitors the bus and automatically performs recovery sequence once timeout is detected. • A manual recovery through a separate I2C channel reset (Link Control register). Refer to the Bus Timeout and Recovery section for more details.
UART	N/A	N/A
Data Channel	If LTPI events are not received by the other end, the APB Data channel stays in its previous state.	None. APB in nature does not have a timeout. It is up to you to terminate the current APB Data channel transaction in case events are not observed within your given timeout/perspective.

7. Hardware Validation

This IP is validated through the LVDS Tunneling Protocol and Interface Reference Design. Refer to the LVDS Tunneling Protocol and Interface (LTPI) (FPGA-RD-02247) for details on the configurations used for validation.

8. Ordering Part Number

The ordering part numbers (OPN) for the DC-SCM LTPI IP Core are listed below.

Table 8.1. Ordering Part Numbers

Device Family	Part Number	
	Single Seat Annual	Single Seat Perpetual
Certus-NX	LTPI-CTNX-US	LTPI-CTNX-UT
CertusPro-NX	LTPI-CPNX-US	LTPI-CPNX-UT
CrossLink-NX	LTPI-CNX-US	LTPI-CNX-UT
Mach-NX	—	LTPI-MNX-UT
MachXO3D	—	LTPI-XO3D-UT
MachXO3	—	LTPI-XO3-UT
MachXO4	LTPI-XO4-US	LTPI-XO4-UT
MachXO5-NX and MachXO5D-NX	—	LTPI-XO5-UT

Appendix A. Resource Utilization

Table A.1 shows resource utilization of DC-SCM LTPI sample configuration for the LCMXO3L device using Synplify Pro of Lattice Diamond software.

Table A.1. Resource Utilization for MachXO3 Device

Configuration	Registers	Slice	LUTs	EBRs
LCMXO3L-9400C-6BG484C *LVDS Clock == 400 MHz* *IP Mode = SCM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 1 (Target)* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1230	731	1434	2
LCMXO3L-9400C-6BG484C *LVDS Clock == 400 MHz* *IP Mode = HPM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 6 (Controller)* *Enable Timer == 111111* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1617	1467	2910	2
LCMXO3L-9400C-6BG484C *LVDS Clock == 400 MHz* *IP Mode = HPM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 6 (Bidirectional)* *Enable Timer == 111111* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1868	1553	3021	2

Note:

- Utilization data is generated using a sample IP configuration for the Lattice Diamond software version 3.14, targeting the LCMXO3L-9400C-6BG484C device with the Strategy set to default setting. Numbers may vary with different software versions, device densities, synthesis tools, or speed grades. For Static Timing Analysis, the IP is tested in both -5 and -6 speed grades. For better performance in certain cases, it is recommended to run multiple iterations of Place and Route and/or set the Optimization Goal to Timing in the Strategy section of the software tool.

Table A.2 shows resource utilization of DC-SCM LTPI sample configuration for the LFMXO5 device using Synplify Pro of Lattice Radiant software.

Table A.2. Resource Utilization for MachXO5-NX Device

Configuration	Registers	Slice	LUTs	EBRs
LFMXO5-25-9BBG400C *LVDS Clock == 600 MHz* *IP Mode = SCM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 1 (Target)* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1323	1299	1788	2
LFMXO5-25-9BBG400C *LVDS Clock == 600 MHz* *IP Mode = HPM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 6 (Controller)* *Enable Timer == 111111* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1742	1708	3260	2
LFMXO5-25-9BBG400C *LVDS Clock == 600 MHz* *IP Mode = HPM* *DDR == Enabled* *Low Latency IO == 16* *Normal Latency IO == 16* *I2C == 6 (Bidirectional)* *Enable Timer == 111111* *UART == 2* *OEM == 16* *Data Channel == Enabled*	1961	1927	3346	2

Note:

- Utilization data is generated using a sample IP configuration for the Lattice Radiant software version 2025.1, targeting the LFMXO5-25-9BBG400C device with the Strategy set to default setting. Numbers may vary with different software versions, device densities, synthesis tools, or speed grades. For better performance in certain cases, it is recommended to run multiple iterations of Place and Route and/or set the Optimization Goal to Timing in the Strategy section of the software tool.

Appendix B. Limitations

The following lists the I2C channel supports:

- I2C tunneling support is limited to Standard (100 kHz) and Fast (400 kHz) modes only.
- Echo support is always enabled.

References

- DC-SCM LTPI IP Core web page
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- Mach-NX web page
- MachXO3 web page
- MachXO3D web page
- MachXO4 web page
- MachXO5-NX web page
- Lattice Radiant Software web page
- Lattice Diamond Software web page
- Lattice Propel Design Environment web page
- Lattice Solutions IP Cores web page
- Open Compute Project web page
- DC-SCM LTPI IP Release Notes (FPGA-RN-02021)
- Lattice Memory Mapped Interface and Lattice Interrupt Interface User Guide (FPGA-UG-02039)
- LVDS Tunneling Protocol and Interface (LTPI) (FPGA-RD-02247)
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

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

Revision 2.1, IP v2.1.0, December 2025

Section	Change Summary
All	• Updated the IP version information on the cover page. • Added a note on the IP version in the *Quick Facts* and *Revision History* sections. • Made editorial fixes.

Introduction	• In Table 1.1. DC-SCM LTPI IP Quick Facts:
	  • Added *MachXO4* to *Supported Devices* and *Lattice Implementation*.
	  • Updated IP and software versions in *Lattice Implementation*.

Ordering Part Number	Added MachXO4 ordering part numbers.
References	Added the *MachXO4* web page.

Revision 2.0, IP v2.0.0, October 2025

Section	Change Summary
All	• Added the OCP logo to the document cover page. • Made editorial fixes.
Abbreviations in This Document	Added *ACK*, *CPLD*, *CSR*, *DDR*, *FIFO*, *FSM*, *GDDR*, *GPIO*, *GUI*, *I/O*, *I2C*, *ID*, *IF*, *IP*, *LL GPIO*, *LSCC*, *LTPI*, *MFR*, *MSB*, *N/A*, *NAK*, *NL*, *NL GPIO*, *OCP*, *OEM*, *OPN*, *PHY*, *Rx*, *SCL*, *SCM*, *SDA*, *SDR*, *SGPIO*, *SMBus*, *Tx*, and *UART*.
Introduction	• Updated the description in the Introduction section. • Updated Table 1.1. DC-SCM LTPI IP Quick Facts. • Updated the DC-SCM version in the Features section.

Functional Descriptions	• Updated the following tables:
	  • Table 2.1. DC-SCM LTPI IP Core Signal Description (including updated Notes)
	  • Table 2.2. Attributes Table for General Tab
	  • Table 2.3. Attributes Description for General Tab
	  • Table 2.4. Attributes Table for Frame Format Tab
	  • Table 2.5. Attributes Description for Frame Format Tab
	  • Table 2.20. DC-SCM LTPI Soft IP Registers
	• Updated the following tables and removed *Tab* from their captions:
	  • Table 2.8. Attributes Table for LL GPIO
	  • Table 2.9. Attributes Description for LL GPIO
	  • Table 2.10. Attributes Table for NL GPIO
	  • Table 2.11. Attributes Description for NL GPIO
	  • Table 2.12. Attributes Table for I2C
	  • Table 2.13. Attributes Description for I2C
	  • Table 2.14. Attributes Table for UART
	  • Table 2.15. Attributes Description for UART
	  • Table 2.16. Attributes Table for OEM
	  • Table 2.17. Attributes Description for OEM
	• Updated the *Corresponding Register* information in the following table:
	  • Table 2.23. Link-Detect Frame Format (including updated Notes)
	  • Table 2.24. Link-Speed Frame Format
	  • Table 2.25. Advertise Frame Format (including updated Notes)
	  • Table 2.32. Configure Frame Format
	  • Table 2.33. Accept Frame Format
	  • Table 2.35. Sample Custom I/O Format
	• Updated *Notes* and *Bit Field* for *Byte Sequence 6* and *7* in Table 2.28. Feature Capability Mapping for Default I/O Frame (Capabilities Type==0x00).

Section	Change Summary

	• Updated *Notes* and *Bit Field* for *Byte Sequence 7* in Table 2.29. Feature Capability Mapping for Custom I/O Frame (Capabilities Type==0x81).
	• Updated descriptions and *Notes* for Table 2.30. Feature Capability Mapping for OEM Defined (Capabilities Type == 0x82-0xFF).
	• Removed the following tables:
	  • *Table 2.6. Attributes Table for Capabilities Tab*
	  • *Table 2.7. Attributes Description for Capabilities Tab*
	• Added the following tables:
	  • Table 2.6. Attributes Table for Data
	  • Table 2.7. Attributes Description for Data
	  • Table 2.18. Summary of DC-SCM LTPI Register Address Space
	• Switched the order of Table 2.19. Access Type Definition and Table 2.20. DC-SCM LTPI Soft IP Registers.
	• Updated the following figures:
	  • Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.5. Frame Format Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.8. FSM of DC-SCM LTPI IP
	  • Figure 2.9. PHY States
	  • Figure 2.10. Sample LTPI Initialization to Active State Timing Diagram
	  • Figure 2.15. Sample Channel Order for Custom I/O
	  • Figure 2.16. Frame Format Tab for Default I/O Frame
	• Removed the following figures:
	  • *Figure 2.6. Capabilities Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.7. LL GPIO Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.8. NL GPIO Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.9. I2C Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.10. UART Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.11. OEM Tab User Interface of DC-SCM LTPI IP Core*
	  • *Figure 2.15. Sample GDDR Synchronization Timing Diagram*
	  • *Figure 2.18. LTPI_VER Byte Mapping*
	  • *Figure 2.19. SPEED_CAP Byte Mapping*
	• Added the following figures:
	  • Figure 2.6. Channels Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.12. Detect Capabilities Local [7:0] Byte Mapping
	  • Figure 2.13. Detect Capabilities Local [23:8] Byte Mapping
	• Updated the following sections:
	  • GDDR Clock Synchronization State
	  • Link Training – Link-Detect State
	  • Link Training – Link-Speed State
	  • Advertise State
	  • Configuration State (for SCM only)
	  • Accept State (for HPM only)
	  • L0 State
	  • Custom I/O Frame
	  • Default I/O Frame
	  • Data Frame
	  • CRC
	  • Word Aligner
	• Added *Note* to the Register Description section.

IP Programming Sequence	Updated this section.

Section	Change Summary

External Channel Interface Handling	• Updated the following tables:
	  • Table 4.2. Sample Customized Payload Configuration
	  • Table 4.6. Summary of Bus Timeout and Recovery Feature
	  • Table 4.8. Data Channel Command Encoding
	  • Table 4.9. Memory Command Frame Mapping
	  • Table 4.11. Summary of Data Channel CRC Handling
	• Updated the descriptions in the following sections:
	  • GPIO Channel
	  • I2C Channel
	  • Bus Timeout and Recovery
	  • Built-in Timer
	  • I2C Channel Reset
	  • UART Channel
	  • OEM Channel
	  • Data Channel
	• Updated the following figures:
	  • Figure 4.6. Bus Recovery with Built-in Timer and Successful STOP Generation in the First Attempt
	  • Figure 4.7. Bus Recovery with Built-in Timer and Automatic Retry Attempts
	  • Figure 4.8. I2C Channel Reset with Manual Reset for LTPI Interfacing with Target Device
	  • Figure 4.9. Channel Reset with Multiple Manual Reset Attempts for LTPI Interfacing with Target Device
	  • Figure 4.17. APB Interface for SCM and HPM
	• Removed the following figures:
	  • *Figure 4.8. I2C Channel Reset with Release on Reset Behavior.*
	  • *Figure 4.11. MCTP over I2C*
	  • *Figure 4.18. Data Channel Pin List*
	  • *Figure 4.20. Example of Data Channel Usage*
	  • *Figure 4.21. Sample SCM to HPM APB Transaction*
	  • *Figure 4.22. Sample HPM IP CSR Access Through Data Channel Block Diagram*
	• Added Figure 4.10. I2C Start and Arbitration Process in Bidirectional I2C.
	• Renamed the *MCTP over I2C* section to Bidirectional I2C and updated its content.

Clocks and Reset	• Updated the descriptions in this section.
	• Updated the following figures:
	  • Figure 5.1. General Clocking Topology (Main)
	  • Figure 5.2. End-to-End Clocking Implementation
	• Removed *Figure 5.2. Suggested Clocking Implementation (Follower)*.
	• Removed the *Clock Compensation* section.
	• Updated the following sections:
	  • Clock Computations
	  • Sample Clock Computations
	  • Reset
	• Added Table 5.1. Recommended System Clock Frequency.

Error Impact, Handling, and Recovery	• Updated the *Rx* information in Table 6.1. IP Error Handling. • Updated the *Recovery* information for *I2C* and *Data Channel* in Table 6.3. Timeout Impact on LTPI Channels.
Ordering Part Number	Updated this section.
Appendix A. Resource Utilization	Updated this section.
References	Added the *DC-SCM LTPI IP Core*, *Certus-NX*, *CertusPro-NX*, *CrossLink-NX*, and *Open Compute Project* web pages.

Revision 1.9, IP v1.6.0, December 2024

Section	Change Summary
All	• Added the IP version information on the cover page. • Updated *I²C* to *I2C*.
Introduction	• Updated the *OCP DC-SCM 2.0 LTPI Specifications* revision and version number in the Features section. • Added *IP Changes* and updated *Lattice Implementation* in Table 1.1. DC-SCM LTPI IP Quick Facts. • Removed *Table 1.2. FPGA Software for IP Configuration, Generation, and Implementation*.
Functional Descriptions	Updated Figure 2.9. FSM of DC-SCM LTPI IP.
External Channel Interface Handling	Updated *Data 0 Echo*, *Data 1 Echo*, and *Data Received Echo* descriptions in Table 4.3. I2C Events Encoding.
References	Added the *Lattice Solutions IP Cores* web page and *DC-SCM LTPI IP Release Notes (FPGA-RN-02021)*.

Revision 1.8, June 2024

Section	Change Summary
All	• Removed *Core* from the document title. • Made editorial fixes.

Abbreviations in This Document	• Replaced *acronyms* with *abbreviations* in this section.
	• Added the following abbreviations:
	  • *Field Programmable Gate Array (FPGA)*
	  • *Management Component Transport Protocol (MCTP)*

Introduction	Updated from *LTPI 1.0 Specifications* to *LTPI 1.1 Specifications*.

Functional Descriptions	• Updated the I²C Channel Interface port details in Table 2.1. DC-SCM LTPI IP Core Signal Description.
	• Updated Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core.
	• Updated content for the following attributes in Table 2.2. Attributes Table for General Tab:
	  • *LTPI Version (Major)*
	  • *LTPI Version (Minor)*
	• Updated Figure 2.9. I2C Tab User Interface of DC-SCM LTPI IP Core.
	• Updated Table 2.12. Attributes Table for I2C Tab:
	  • Added *Number of MCTP bus interface* and *Total number of Internal I2C buses used* attributes.
	  • Updated *Dependency on Other Attributes* content for the *I2C Bus interface per Frame*, *Enable Timer for Bus #*, and *Generate STOP on Bus # Reset* attributes.
	• Updated the content in the CRC section.

External Channel Interface Handling	Added MCTP over I2C section and updated the remaining figure numbers in this section accordingly.
References	Updated this section and added the following references: • Lattice Radiant Software web page • Lattice Diamond Software web page • Lattice Propel Design Environment web page • LVDS Tunneling Protocol and Interface (LTPI) (FPGA-RD-02247)

Revision 1.7, November 2023

Section	Change Summary
Disclaimers	Updated this section.
Inclusive Language	Newly added this section.

Functional Descriptions	• Table 2.1. DC-SCM LTPI IP Core Signal Description:
	  • changed the Port Name *lnit_done_o* to *lnk_init_done_o* under Others;
	  • newly added *i2c_tmr_to_o* and *i2c_code_err_o* ports under Others.
	• Updated Figure 2.6. Capabilities Tab User Interface of DC-SCM LTPI IP Core.
	• Table 2.6. Attributes Table for Capabilities Tab:
	  • changed the Default value of the Default Feature Capability 0 in Hex (0x) attribute from *32'h7f00201f* to *32'h7f00101f*.

Section	Change Summary

	• Updated Figure 2.8. NL GPIO Tab User Interface of DC-SCM LTPI IP Core.
	• Table 2.10. Attributes Table for NL GPIO Tab:
	  • newly added the Number of NL GPIO attribute;
	  • updated Selectable Values and Dependency on Other Attributes for NL GPIO Input Data Width and NL GPIO Output Data Width attributes;
	  • updated the Dependency on Other Attributes for the NL GPIO Payload Width per Frame attribute;
	  • added the following table note:
	  NL GPIO input and output data width are always equal to Number of NL GPIO attribute. It is up to you to map the virtual I/O to the targeted input and output physical pins.
	• Table 2.11. Attributes Description for NL GPIO Tab:
	  • newly added the Number of NL GPIO attribute;
	  • updated Description of NL GPIO Input Data Width, NL GPIO Output Data Width, and NL GPIO Payload Width per Frame attributes.
	• Updated Figure 2.9. I2C Tab User Interface of DC-SCM LTPI IP Core.
	• Table 2.12. Attributes Table for I2C Tab and Table 2.13. Attributes Description for I2C Tab:
	  • removed the original Timing section and added the Bus # section;
	  • placed the following attributes under Bus # and updated corresponding descriptions:
	  I2C Bus # Mode, Enable Fast Mode for Bus #, Customize Bus # Tsu-sda, Bus # Tsu-sda count, Enable Timer for Bus #, Disable Timer Prescaler for Bus #, Prescaler Ratio for Bus #, Timer Size of Bus #, Apply Bus 0 Timer Settings to all I2C Buses, Generate STOP on Bus # Reset.
	• Table 2.18. DC-SCM LTPI Soft IP Registers:
	  • updated Field Description of i2c_bus_rst in the I2C_BUS_RST register;
	  • in the Bit column, changed *[31:11]* to *[31:13]* for RSVD in the INT_STATUS register;
	  • newly added i2c_code_err_int and i2c_tmr_to_int in the INT_STATUS register;
	  • in the Bit column, changed *[31:11]* to *[31:13]* for RSVD in the INT_ENABLE register;
	  • newly added i2c_code_err_en and i2c_tmr_to_en in the INT_ENABLE register;
	  • in the Bit column, changed *[31:11]* to *[31:13]* for RSVD in the INT_SET register;
	  • newly added i2c_code_err_set and i2c_tmr_to_set in the INT_SET register;
	  • added – *set* to Field Description of acpt_to_set, cfg_to_set, and ls_to_set.
	• Table 2.27. Feature Capability Mapping for Default I/O Frame (Capabilities Type==0x00) and Table 2.28. Feature Capability Mapping for Custom I/O Frame (Capabilities Type==0x81):
	  • in the Bit Field column, changed Total Number of NL GPIO[7:0] to Number of NL GPIO[7:0] and changed Total Number of NL GPIO[9:8] to Number of NL GPIO[9:8];
	  • updated the content of Note 1 to total Number of NL GPIO that is tunneled over LTPIO, the Number of NL GPIO attribute.
	• In the CRC section, added CRC is computed using inverted/reflected input and output, that is in the stream of bytes, the least significant bit of each byte is processed first and the resulting CRC is inverted to get the final CRC value.

External Channel Interface Handling	• Changed Selectable Values from 16 to 64 for the NL GPIO Output Data Width attribute in
	• Table 4.2. Sample Customized Payload Configuration.
	• In the GPIO Channel section:
	  • added NL GPIO Input Data Width is always equal to NL GPIO Output Data Width to maintain the same latency in both directions;
	  • added it is up to the system level design to map the virtual GPIOs to actual input and output physical ports;
	  • removed depending on NL GPIO Output Data Width, IP switches distribution between each 8-bit GPIO output pins until it can wrap around the total pins;
	  • changed NL Frame Counter takes non-zero value that wraps around every M count to
	  NL Frame Counter starts at count 0 that wraps around every M-1 count and updated the corresponding equation;
	  • changed when NL PGIO is disabled, frame counter is fixed to 1 to
	  when NL GPIO is disabled, frame counter is fixed to 0.

Section	Change Summary
	• changed IP samples the NL GPIO input ports based on NL GPIO Payload Width per Frame attribute set and multiplexes sampling around the total NL GPIO Input Data Width to IP samples the virtual GPIO based on NL GPIO Payload Width per Frame attribute set and multiplexes sampling around the total Number of NL GPIO set; • changed index sampling is determined by NL Frame Counter which occupies the third byte of Default I/O frame and 15th byte of Custom I/O frame to index sampling is determined by NL Frame Counter which occupies the third byte of Default I/O frame and is customizable in the Custom I/O frame; • changed if there is a total of 64 NL GPIO Input Data, IP switches sampling between each 8-bit GPIO input pins per sampling time with NL Frame Counter starting at 1, until it can wrap around the total input pins with NL Frame Counter wrapping around at 8 to if there is a total of 64 Number of NL GPIO, IP switches sampling between each 8-bit virtual GPIO input pins per sampling time with NL Frame Counter starting at 0, until it can wrap around the total input pins with NL Frame Counter wrapping around at 7; • changed for the received frames, IP remaps the received NL GPIO payload into the GPIO output pins to for the received frames, IP remaps the received NL GPIO payload into the virtual GPIO output pins. • Updated Figure 4.2. NL GPIO Mechanism. • Updated Figure 4.4. I2C Mechanism. • Table 4.3. I2C Events Encoding: • removed *0b1000* from Description of Reserved; • changed 0b1100-0b1111 to 0b1101-0b1111. • Added the below paragraph in the I²C Channel section: In any case that the LTPI controller or target detects invalid sequence of events, i2c_code_err_o asserts to indicate error status. When this flag is asserted, I²C bus may encounter hang-up or timeout depending on when the error is detected. The bus can be recovered by triggering I²C channel reset and/or using the built-in timer. If the built-in timer is enabled, IP performs automatic recovery when error is encountered. If the built-in timer is not enabled, you must perform I²C channel reset to recover the bus during hang-up. Refer to the Bus Timeout and Recovery section for more details. • Added the Bus Timeout and Recovery section. • Updated Figure 4.14. UART Mechanism. • Updated Figure 4.16. OEM Mechanism.
Error Impact, Handling, and Recovery	Updated Recovery description of the I²C channel in Table 6.3. Timeout Impact on LTPI Channels.
Ordering Part Number	Updated the License Type to *Multi-site Perpetual* for the following OPNs: LTPI-XO3D-UT, LTPI-XO3-UT, LTPI-MNX-UT, and LTPI-XO5-UT.
Appendix A. Resource Utilization	Updated Table A.1. Resource Utilization for MachXO3 and Table A.2. Resource Utilization for MachXO5-NX.
References	Newly added links to MachXO3D Family Devices Web Page and Lattice Insights for Training Series and Learning Plans.

Revision 1.6, June 2023

Section	Change Summary
Introduction	Deleted paragraphs *DC-SCM aims to move common server management, security and control features from a typical motherboard into a module designed in different form factors (horizontal, vertical). The basic idea is to enable a common security and management module form and interface, which can be used across datacenter platforms.* *From a Data Center perspective, DC-SCM enables a common management and security to be deployed across a higher percentage of platforms. It also enables deployment of management and security upgrades on platforms within a generation without redesign of more complex components. From development perspective, this enables solution providers to remove customer specific solutions from the more complex components (such as motherboards). This enables greater leverage of higher complexity components across platforms.* *Also, the DC-SCM LTPI IP Core provides a solution for minimal wire connection between two FPGAs to provide TDM-based bidirectional communication, aggregating multiple data such as I2C, GPIO and UART*

Section	Change Summary
	to add more flexibility to a customer's system and board design. This solution is compliant with Datacenter-ready Secure Control Module (DC-SCM) 2.0.
Functional Descriptions	• Deleted sys_clk_o and updated sync_rdy_o port information in Table 2.1. DC-SCM LTPI IP Core Signal Description. • Updated Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core and Figure 2.20. Target Speed Processing Illustration. • Added Table Note 2 and 3 information in Table 2.6. Attributes Table for Capabilities Tab. • Added When IP is meant to interface with an external I2C controller, I2C Bus Mode must be set to Controller. When IP is meant to interface with an external I2C target, I2C Bus Mode must be set to Target in I2C Bus # Mode description in Table 2.13. Attributes Description for I2C Tab. • Added paragraph In Figure 2.14 is the high-level illustration of the states of the LTPI IP, starting from initialization up to active state: 1) DDR synchronization (GDDR Clock Synchronization State), 2) Link training and speed negotiation (Link Training – Link-Detect State and Link Training – Link-Speed State), 3) DDR synchronization, 4) Feature negotiation (Advertise State and Configuration State (for SCM only) or Accept State (for HPM only)), and Active state (L0 State) in PHY states section. • Added paragraph GDDR synchronization is required by the DDR interface to properly synchronize DDR related clocks. It must be started after DDR related clocks in the system become stable (for example, sync_clk_i, eclk_i, eclk90_i, clk_i, etc.). GDDR clock synchronization logic is always clocked by sync_clk_i in GDDR Clock Synchronization State section. • Added Figure 2.15. Sample GDDR Synchronization Timing Diagram, Figure 2.21. Sample Channel Order for Custom I/O, and Figure 2.22. Frame Format Tab for Default I/O Frame. • Added Timing Diagram section, Figure 2.16. Sample LTPI Initialization to Active State Timing Diagram, and Table 2.21. Summary of Start of Frame and Frame Sub-type. • Added sentence They are consist of LTPI version and speed capability of the LTPI IP, and are later on used to determine the common target speed between SCM and HPM in Link-Detect Frame section. • Added Table note information in Table 2.22 Link-Detect Frame Format and Table 2.29. Feature Capability Mapping for OEM Defined (Capabilities Type==0x82-0xFF). • Added paragraph When Capabilities Type==0x82-0xFF (OEM defined) is used, feature capability is solely based on user-defined value and mapping is shown in Table 2.29. Definition of actual feature mapping for OEM defined capability type is out of scope of the soft IP. For the external channels, IP uses the default feature settings set during IP generation (for example, I2C channel follows the I2C bus speed set during IP generation (100 kHz if Enable Fast Mode for Bus == Unchecked and 400 kHz if Enable Fast Mode for Bus == Checked) in Advertise Frame section. • Added paragraph Target speed is generated from the speed capability information in Link-Detect frames. It is determined based on the fastest clock capability that is common between SCM and HPM. In the example in Figure 2.20, speed capability of SCM Link-Detect frames supports 200 MHz, 100 MHz, and 25 MHz DDR clocks while HPM supports 100 MHz and 25 MHz DDR clocks. The fastest common speed between the two is 100 MHz DDR. The IP uses this as the target speed for Link-Speed Frame in Link-Speed Frame section. • Added Table Note information in Table 2.24. Advertise Frame Format.
IP Programming Sequence	Added sentence *User must make sure that IPs are configured in such a way that SCM and HPM will have a matching feature when attribute Automatically move to Configuration State is enabled. There is no way to manually reconfigure the requested features when this attribute is set* in Request Features section.
External Channel Interface Handling	Replaced *Due to implementation of glitch filter, minimum of 40 MHz input system clock is required for IP with I2C channel enabled to work correctly. When I²C channel is enabled, IP must be configured to a setting that has equivalent input system clock that is greater than or equal to 40 MHz (example given 400 MHz LVDS clock). IP with I2C channel enabled is not guaranteed to work if input system clock is less than 40 MHz* with *Due to implementation of glitch filter and nature of I2C tunneling logic, minimum of 10 MHz input system clock is required for IP with I²C 100 kHz channel enabled and 20 MHz input system clock for I2C 400 kHz channel enabled to work correctly (for example, 50 MHz DDR or 100 MHz SDR LVDS clock and above for I2C 100 kHz mode, 100 MHz DDR or 200 MHz SDR LVDs clock and above for I2C 400 kHz mode). IP with I2C channel enabled is not guaranteed to work if input system clock frequency is less than the minimum required for each mode* in Glitch Filter section.
Clocks and Reset	• Added bullet information APB clock (apb_pclk_i) is used for IP CSR access. When Data Channel is enabled, this clock is also used by the APB interface of the Data Channel and frequency is recommended to be equal or less than the system clock (clk_i).

Section	Change Summary
	• Deleted bullet information GDDR Rx generates core clock (sclk) based on received LVDS Rx clock from LTPI. All CDC transfers in the DDR domain are already handled by the hard IPs instantiated to implement the DDR clock tree. • Added paragraph Clock topology implementation is outside of scope of the IP. All clocks, aside from LVDS Tx clock (lvds_tx_clk_o) and core clock (sclk r/tx), are input to the IP and correct clock switching, which is required by DC-SCM LTPI 1.0, must be done externally which includes but is not limited to reconfiguration of GPLL/clock source that clocks the DDR interface. Refer to Clock Switching section for details. • Deleted paragraph The IP only supports external clock switching from base to target frequency. Clock implementation is outside of scope of the IP. No PLL is used inside the IP, and it is up to the user to correctly configure the clock. DC-SCM LTPI 2.0 requires dynamic switching of clock from base frequency to target frequency. This requires reconfiguration of GPLL/clock source that clocks the DDR interface. Refer to Clock Switching section for details. • Added Figure 5.3. End to End Clocking Implementation. • Added Clock section and deleted Figure 5.3. End to End Clocking Implementation.

Revision 1.5, March 2023

Section	Change Summary
Introduction	In the Features section, revised the feature for the I²C interface to *each can be configured as Controller or Target*.

Functional Descriptions	• Updated Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core.
	• In Table 2.2. Attributes Table for General Tab:
	  • stated that LTPI Version (Major) and LTPI Version (Minor) are *uneditable* in the Dependency on Other Attributes column;
	  • changed the default value for LTPI Version (Minor) to *0*;
	  • changed the default value for LTPI Version to *1.0*.

Hardware Validation	Added the link to the LVDS Tunneling Protocol and Interface Reference Design webpage.

Revision 1.4, February 2023

Section	Change Summary
Introduction	Added reference to Table A.2. Resource Utilization for MachXO5-NX in Table 1.1. DC-SCM LTPI IP Quick Facts.

Functional Descriptions	• Updated Table 2.1. DC-SCM LTPI IP Core Signal Description.
	  • Updated soft_reset_i, apb_pclk_i, and apb_pslverr_o description.
	  • Updated headers to APB Completer Interface (IP CSR) and APB Requester Interface (Data Channel).
	  • Added APB Completer Interface (Data Channel) group.
	• Updated user interface screenshots in the Attribute Summary section.
	• Updated default values in Table 2.4. Attributes Table for Frame Format Tab.
	• Updated the introduction statement of the Register Description section.
	• Removed the Register Address Space figure.
	• Updated Figure 4.3. I2C Pin List and description.
	• Updated Figure 4.4. I2C Mechanism and description.
	• Updated Figure 4.20. Sample SCM to HPM APB Transaction and used APB_C_READY in the description.

Hardware Validation	Added this section.
Ordering Part Number	Added this section.
Appendix A. Resource Utilization	• Updated values in in Table A.1. Resource Utilization for MachXO3. • Added Table A.2. Resource Utilization for MachXO5-NX.
References	Added references to the Mach-NX and MachXO5-NX web pages in latticesemi.com.
All	• Updated structure of information.

Section	Change Summary
	• Corrected product names and trademarks. • Corrected links to tables. • Minor adjustments in formatting and style.

Revision 1.3, December 2022

Section	Change Summary

Introduction	• In Table 1.1. DC-SCM LTPI IP Quick Facts:
	  • Added MachXO5-NX family to the Supported FPGA Families
	  • Added LFMXO5to the Targeted Devices.
	• Added Table 1.2. FPGA Software for IP Configuration, Generation, and Implementation.
	• In the Features section:
	  • Added sub LVDS.
	  • Add new bullet Supports up to 1200 Mbps LVDS data rate for MachXO5-NX family devices.

Functional Descriptions	• Updated Table 2.1. DC-SCM LTPI IP Core Signal Description, Table 2.2. Attributes Table for General Tab, and Table 2.24. Platform Field Bit Mapping.
	• Added Table 2.3. Attributes Description for General Tab, Table 2.4. Attributes Table for Frame Format Tab, and Table 2.5. Attributes Description for Frame Format Tab.
	• Updated below figures:
	  • Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.6. Capabilities Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.7. LL GPIO Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.8. NL GPIO Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.9. I2C Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.10. UART Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.11. OEM Tab User Interface of DC-SCM LTPI IP Core
	  • Figure 2.40. General Clocking Topology (Main)
	  • Figure 2.41. Suggested Clocking Implementation (Follower)
	• Added Figure 2.5. Frame Format Tab User Interface of DC-SCM LTPI IP Core.
	• Updated Custom I/O Frame in the Data I/O Frames section.
	• Added the sentence Payload byte offset allocation for each channel is indicated by Frame Offset Byte # attribute. If any of the channels is disabled, the corresponding allocated payload field is padded with 1s in Default I/O Frame in the Data I/O Frames section.
	• Updated the sentence from IP with I2C channel enabled does not work if input system clock is less than 40 MHz to IP with I2C channel enabled is not guaranteed to work if input system clock is less than 40 MHz in the Glitch Filter section.
	• Updated sentence from It follows the sampling mechanism similar to LL GPIO except that when enabled, OEM interface is updated only every non-Data Channel frame transfer to It follows sampling mechanism similar to LL GPIO except that when enabled, OEM pins are automatically mapped to IO buffers and interface is updated only every non-Data Channel frame transfer in the OEM Channel section.
	• Added the sentence The approach mentioned is just a recommendation and actual clock implementation is outside the scope of the IP in the Clocks and Reset section.

Revision 1.2, September 2022

Section	Change Summary

Introduction	• In Table 1.1. DC-SCM LTPI IP Quick Facts:
	  • added *Mach-NX* to the Supported FPGA Families;
	  • added LFMNX to the Targeted Devices.
	• In the Features Section:
	  • added *Mach-NX* to the devices that support up to 800 Mbps LVDS data rate.

Functional Descriptions	• Rearranged the original Table 2.2. Attributes Table into the following tables according to available

Section	Change Summary
	tabs in the GUI: Table 2.2. Attributes Table for General Tab, Table 2.4. Attributes Table for Capabilities Tab; Table 2.6. Attributes Table for LL GPIO Tab; Table 2.8. Attributes Table for NL GPIO Tab; Table 2.10. Attributes Table for I2C Tab; Table 2.12. Attributes Table for UART Tab; Table 2.14. Attributes Table for OEM Tab. • Rearranged the original Table 2.3. Attributes Description into the following tables according to available tabs in the GUI: Table 2.3. Attributes Description for General Tab; Table 2.5. Attributes Description for Capabilities Tab; Table 2.7. Attributes Description for LL GPIO Tab; Table 2.9. Attributes Description for NL GPIO Tab; Table 2.11. Attributes Description for I2C Tab; Table 2.13. Attributes Description for UART Tab; Table 2.15. Attributes Description for OEM Tab. • Newly added Figure 2.4. General Tab User Interface of DC-SCM LTPI IP Core; Figure 2.5. Capabilities Tab User Interface of DC-SCM LTPI IP Core; Figure 2.6. LL GPIO Tab User Interface of DC-SCM LTPI IP Core; Figure 2.7. NL GPIO Tab User Interface of DC-SCM LTPI IP Core; Figure 2.8. I2C Tab User Interface of DC-SCM LTPI IP Core; Figure 2.9. UART Tab User Interface of DC-SCM LTPI IP Core; Figure 2.10. OEM Tab User Interface of DC-SCM LTPI IP Core. • Updated the Description for the Target System Clock (MHz) Attribute in Table 2.5. Attributes Description for Capabilities Tab. • Updated the Description for the LL GPIO Input Data Width Attribute in Table 2.7. Attributes Description for LL GPIO Tab. • Updated the Description for the NL GPIO Input Data Width Attribute in Table 2.9. Attributes Description for NL GPIO Tab. • Updated the Description for the Number of I2C bus interface Attribute in Table 2.11. Attributes Description for I2C Tab. • Updated the Description for the Number of UART bus interface Attribute in Table 2.13. Attributes Description for UART Tab. • Updated the Description for the OEM Data Width Attribute in Table 2.15. Attributes Description for OEM Tab.

Revision 1.1, August 2022

Section	Change Summary
All	Minor changes in formatting across the document.

Functional Description	• Updated the following in Table 2.2. Attributes Table:
	  • Removed *Data Frame Type == Custom* from the condition in some of the attributes and added two rows for new attributes in the I^2C group.
	  • Removed *Data Frame Type == Custom* from the condition in some of the attributes in the UART group.
	• Updated Table 2.3. Attributes Description to add two rows for new attributes.
	• Updated Table 2.18. DC-SCM LTPI Soft IP Registers to change default value of 0x64 to 24'h0.
	• Updated Figure 2.13. FSM of DC-SCM LTPI IP.

Revision 1.0, June 2022

Section	Change Summary
All	Initial release.

www.latticesemi.com