DC-SCM LTPI IP Core - Lattice Propel Builder

User Guide

FPGA-IPUG-02200-1.1

August 2022

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 associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
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)
CRC	Cyclic Redundancy Check
DC-SCI	Datacenter-ready Secure Control Interface
DC-SCM	Datacenter-ready Secure Control Module
GUID	Globally Unique Identifier
HEC	Header Error Control
HPM	Host Processor Module
HPM FPGA	Host Processor Module FPGA
LVDS	Low Voltage Differential Signaling
LTPI	LVDS Tunneling Protocol and Interface
MCSI	Multi-Channel Serial Interface
TDM	Time Division Multiplexing

1. Introduction

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

LTPI is a protocol and interface designed for tunneling various low-speed signals between HPM and 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 as the replacement to 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 I^{2}C and UART. It is also extensible with additional proprietary OEM interfaces and provides support for raw Data tunneling between HPM CPLD and SCM CPLD.

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 I^{2}C, GPIO and UART 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.

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 FPGA Families	MachXO3L™, MachXO3LF™, MachXO3D™
Resource Utilization	Targeted Devices	LCMXO3L, LCMXO3LF, LCMXO3D
	Supported User Interface	GPIO, I^{2}C, UART, OEM, Data to LVDS and vice versa
	Resources	See Table A.1.
Design Tool Support	Lattice Implementation	IP Core v1.1.x – Lattice Propel™ Builder 2.1
	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 guide.

1.2. Features

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

- Compliant with OCP DC-SCM 2.0 LTPI 1.0 Specifications
- Link initialization, discovery and negotiation.
- Supports Multi-Channel Serial Interface
- Supports LVDS
- Supports up to 5 channels aggregation/disaggregation in total
- Supports GPIO, I^{2}C, UART, OEM, and Data channel aggregation
- For I^{2}C interface, each can be configured as Controller, Target or Controller/Target (for multi-controller/main).
- Supports up to 800 Mbps LVDS data rate for MachXO3™ 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 DC-SCM LTPI IP is shown in Figure 2.1. Data received from external channels are aggregated and transmitted from SCM side to Host Processor Module (HPM) through Low Voltage Differential Signaling (LVDS). Data received from HPM is de-aggregated and remapped to external channels.

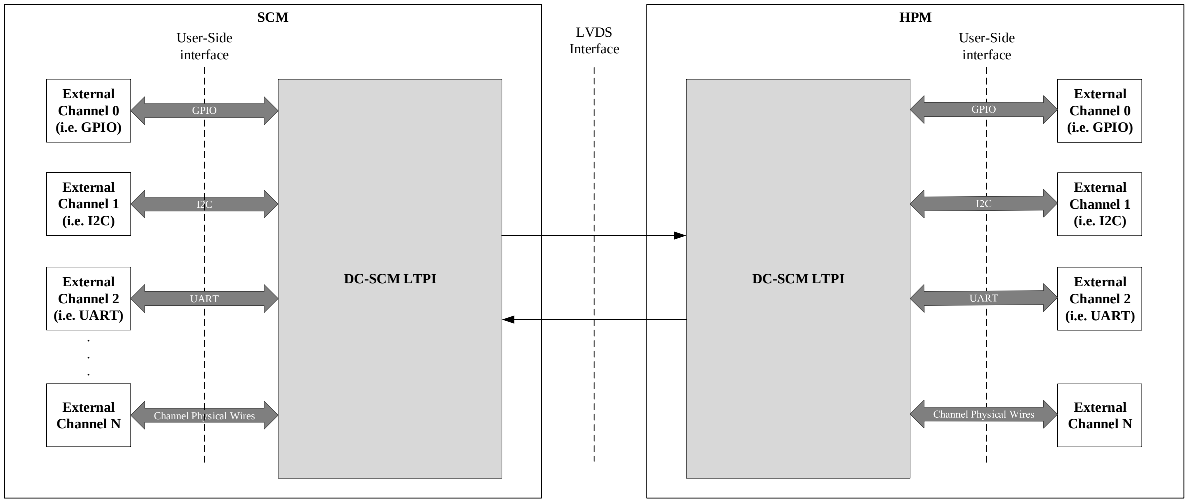


Figure 2.1. High-Level Block Diagram

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

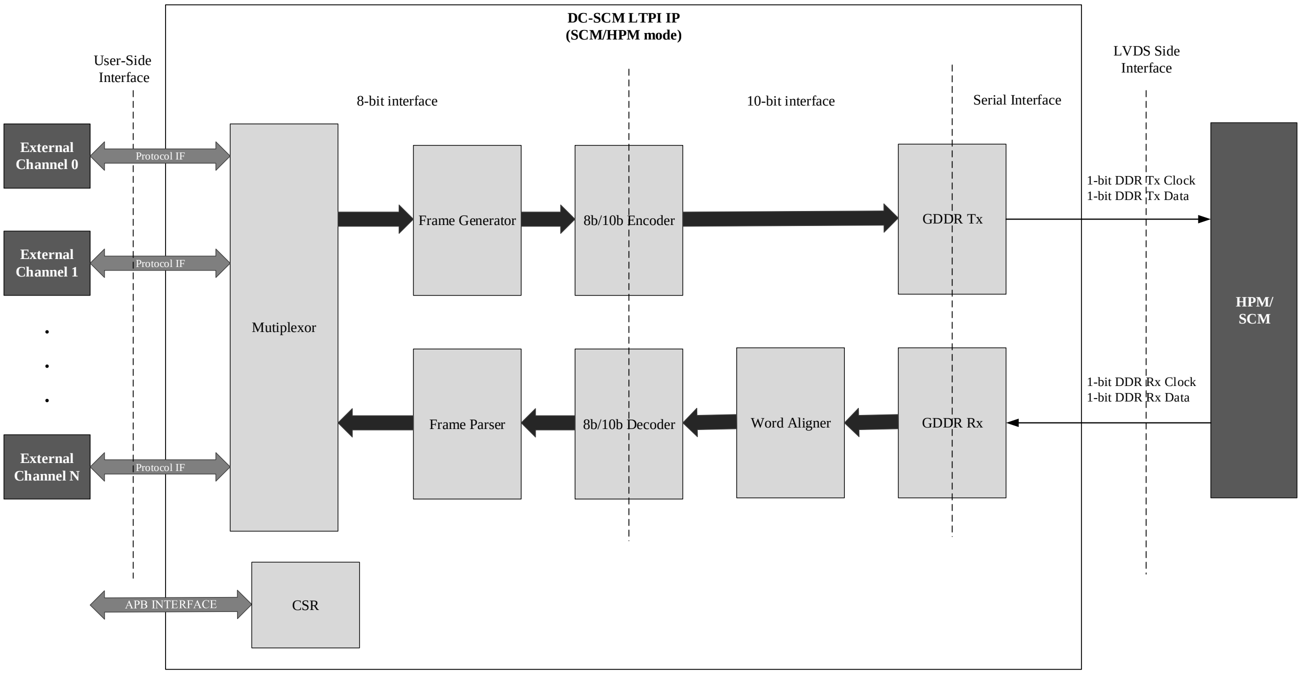


Figure 2.2. Functional Block Diagram of DC-SCM LTPI set as either SCM or HPM

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

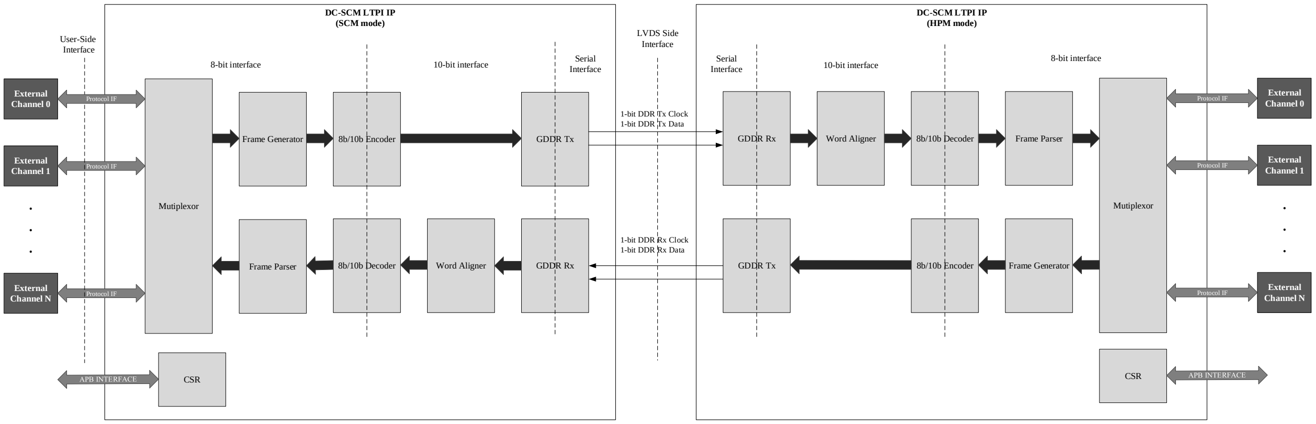


Figure 2.3. DC-SCM LTPI End-to-End SCM to HPM Connection Functional Block Diagram

DC-SCM/HPM LTPI IP is consists of Multiplexor, Frame Generator/Parser, 8b/10b Encoder/Decoder, Word Aligner/Link Synchronizer and GDDR Transmit and Receive modules. In transmit mode, 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 this IP has both Rx and Tx paths available that work simultaneously.

Data received from the user-side are considered valid data streams for transmit. Payload within frames received are considered valid.

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. Same frequency as LVDS Clock (MHz)
eclk90_i	I	1	N/A	90-degree shifted for transmit clock generation. Same frequency as LVDS Clock (MHz) and must be source synchronous with eclk_i.
sync_clk_i	I	1	N/A	GDDR_SYNC low speed continuously running clock input. Must be slower or equal than the slowest clock in the system.
sync_rst_i	I	1	N/A	Active-high GDDR synchronization reset.
sync_start_i	I	1	N/A	Start the GDDR sync process of both PHY Tx and Rx. Must be asserted only if all clocks (Rx and Tx) are already stable. Must be HIGH during all synchronization process. Can be tied to GPLL lock of source synchronous clocks

clk_i	I	1	N/A	Input system clock used to clock both Tx and Rx logic.
				Frequency should be equivalent to *LVDS Clock (MHz) /* PHY_MODE
				PHY_MODE = 5 (for DDR)
				PHY_MODE = 10 (for SDR)

reset_n_i	I	1	N/A	Active-low system reset

sys_clk_o	O	1	N/A	Output Rx SCLK generated by the IP. This is based on DDR system used.
				Frequency is equivalent to *LVDS Clock (MHz)/ (DDR_GEAR/DDR_MODE)*
				DDR_GEAR = 8 (MachXO3) ;
				DDR_MODE = 2 (for DDR) ; DDR_MODE = 1 (for SDR)

soft_reset_i	I	1	N/A	Soft-reset of the system

Port Name	I/O	Width	Default Value	Description
				Reset only applies to interface controller and Link-training and negotiation FSM-related logic.
start_tx_i	I	1	N/A	External control to start Tx path process. When asserted, IP starts link-training and negotiations. Must be HIGH during all IP process. If deasserted, system reset must also be performed.
start_rx_i	I	1	N/A	External control to start Rx path process. When asserted, IP starts link-training and negotiations. Must be HIGH during all IP process. If deasserted, system reset must also be performed.
int_o	O	1	1'b0	Interrupt signal
Protocol Information[8]				
tgt_spd_o	O	16	16'h0	Indicates the target speed of the link
tgt_spd_vld_o	O	1	1'b0	Indicates valid target speed of the link
rx_feat_cap_o	O	64	64'h0	Received feature capability
rx_feat_cap_vld_o	O	1	1'b0	Received feature capability valid
clkcfg_en_o	O	1	1'b0	Indicates clock can be reconfigured for target frequency. This is asserted the first time IP goes to Advertise state.
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[1]				
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[2]				
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
I^{2}C Channel Interface[3]				
i2c_scl_io	I/O	Number of I^{2}C bus interface	N/A	I^{2}C SCL (clock) pin
i2c_sda_io	I/O	Number of I^{2}C bus interface	N/A	I^{2}C SDA (data) pin
UART Channel Interface[4]				
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	{*Number of*	UART Channel output pin

Port Name	I/O	Width	Default Value	Description
		UART bus interface	*UART bus interface*{1'b1}}	
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[5]				
oem_io	I/O	OEM Data Width	N/A	OEM pins
oem_output_en_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.
Data Channel Interface[6]				
data_ch_i	I	8	N/A	Data Channel miscellaneous input pins. Allocated to "Tag" field in Data Frames.
data_ch_o	O	8	{8{1'b1}}	Data Channel miscellaneous output pins. Allocated to "Tag" field in Data Frames.
APB Completer Interface				
apb_pclk_i	I	1	N/A	APB clock. Frequency is recommended to be the same as clk_i. When *Enable Data Channel == Enabled,* this clock is also used by the APB Requester logic of HPM IP.
apb_reset_n_i	I	1	N/A	APB active low reset.
apb_psel_i	I	1	N/A	Select signal. 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. Indicates the second and subsequent cycles of an APB transfer.
apb_pready_o	O	1	1'b0	Ready signal. 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. Only valid for Data Channel related transactions to indicate CRC error or completion error in Data Frames. Not supported and just tied to 0 for IP CSR related transactions.
APB Requester Interface[9]				
apb_r_prdata_i	I	32	N/A	Read data from APB completer device.
apb_r_pready_i	I	1	N/A	Ready signal. Indicates transfer completion.
apb_r_pslverr_i	I	1	N/A	APB error signal. Error signal indicating failure of a transfer.
apb_r_penable_o	O	1	1'b0	Enable signal.
apb_r_psel_o	O	1	1'b0	Select signal.
apb_r_paddr_o	O	32	32'h0	Address signal.
apb_r_pwrite_o	O	1	1'b0	Direction signal. Write = 1, Read = 0
apb_r_pwdata_o	O	32	32'h0	Write data signal.
dc_err_o	O	1	1'b0	Flag signal indicating that HPM received CRC Error Data Frame from remote IP (SCM). This means that previous completion data frame transaction from HPM is not received correctly by SCM.
Others[7]				
rx_align_stat_o	O	1	1'b0	Active HIGH signal that indicates Rx word alignment is achieved.
crc_err_o	O	1	1'b0	Indicates that CRC error is detected in the received frame.
init_done_o	O	1	1'b0	Indicates that LTPI link training and negotiation is done.

Port Name	I/O	Width	Default Value	Description
link_err_o	O	1	1'b0	Indicates that IP encountered link error. When asserted, it generally means that Link-Training must be performed again. Refer to Figure 2.6 for more details.
sync_ready_o	O	1	1'b0	Indicates that GDDR sync startup is finished and Tx/Rx circuit are ready to operate. This can toggle multiple times during Rx alignment until alignment is achieved.
sample_done_o	O	1	1'b0	Indicates that current payload is sampled.
ls_to_o	O	1	1'b0	Indicates that timeout is encountered during Link-Speed state. Once asserted, this is only unset when IP enters the GDDR synchronization again.
cfg_to_o	O	1	1'b0	Indicates that timeout is encountered during Configuration state when IP is set as SCM. Once asserted, it is only unset when IP enters Configuration state again or Link-Training is restarted.
acpt_to_o	O	1	1'b0	Indicates that timeout is encountered during Accept state when IP is set as HPM. Once asserted, this is only unset when IP enters Accept state again or Link-Training is restarted.

Notes:

1. Available only if *Enable Low Latency GPIO Channel==Enabled*.
2. Available only if Enable Normal Latency GPIO Channel==Enabled.
3. Available only if Enable I^{2}C Channel==Enabled.
4. Available only if *Enable UART Channel==Enabled*.
5. Available only if *Enable OEM Channel==Enabled*.
6. Available only if *Enable Data Channel==Enabled*.
7. Available only if *Enable Miscellaneous Signals==Enabled*.
8. Available only if *Enable Protocol Information Signals == Enabled*. These information can be used when programming the IP.
9. Available only if *Enable Data Channel==Enabled* and *IP Mode==HPM*.

2.3. Attribute Summary

The configurable attributes of the DC-SCM LTPI IP Core are shown in Table 2.2 and are described in Table 2.3. The attributes can be configured through the IP Catalog's Module/IP wizard of the Lattice Radiant software.

Table 2.2. Attributes Table

Attribute	Selectable Values	Default	Dependency on Other Attributes
General			
ID Number in Hex (0x)	—	16'h0	—
IP Mode	SCM, HPM	SCM	—
LTPI Version (Major)	0–15	1	—
LTPI Version (Minor)	0–15	1	—
LTPI Version	—	1.1	*LTPI Version (Major). LTPI Version (Minor)*
I/O Type	LVDS	LVDS	—
Customize CRC Polynomial	Checked, Unchecked	Unchecked	
CRC-8 Polynomial in Hex	—	8'h07	Editable only if *Customize CRC Polynomial* == Checked
CRC-8 Initial Value	—	8'h00	Editable only if *Customize CRC Polynomial* == Checked
Enable Miscellaneous Signals	Enabled, Disabled	Disabled	—
Enable Protocol Information Signals	Enabled, Disabled	Disabled	—
Enable Clock Compensation	Checked, Unchecked	Unchecked	—
Frame Settings			
Data Frame Type	Custom, Default I/O	Default I/O	—

Attribute	Selectable Values	Default	Dependency on Other Attributes
Enable Low Latency GPIO Channel	Enabled, Disabled	Enabled	Editable only if *Data Frame Type* == Custom
Enable Normal Latency GPIO Channel	Enabled, Disabled	Enabled	Editable only if *Data Frame Type* == Custom
Enable I^{2}C Channel	Enabled, Disabled	Enabled	Editable only if *Data Frame Type* == Custom
Enable UART Channel	Enabled, Disabled	Enabled	Editable only if *Data Frame Type* == Custom
Enable OEM Channel	Enabled, Disabled	Enabled	Editable only if *Data Frame Type* == Custom
Total Number of Payload Bytes	1–12	12	*(LL GPIO Payload Width per Frame* + *NL GPIO Payload Width per Frame* + *I^{2}C Bus interface per Frame* × 4 + *UART Bus interface per Frame* × 4 + *OEM Payload Width per Frame)* Uneditable Total payload must be 12 bytes max
Enable Data Channel	Enabled, Disabled	Enabled	—
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)[1]	Checked, Unchecked	Unchecked	—
X24 (600 MHz)	Checked, Unchecked	Unchecked	Uneditable
X32 (800 MHz)	Checked, Unchecked	Unchecked	Uneditable
X40 (1000 MHz)	Checked, Unchecked	Unchecked	Uneditable
Target System Clock (MHz)	—	2.5	Uneditable
Feature Capability			
Enable Full OEM Capabilities Type	Checked, Unchecked	Unchecked	Editable only if *Data Frame Type* == Custom
Type value in Hex (0x)	—	8'h81	Viewable and editable only if *Enable Full OEM Capabilities Type* == Checked
Capabilities Type	—	8'h00	Uneditable
Default Feature Capability 0 in Hex (0x)	—	32'h7f00201f	Uneditable Viewable only if *Enable Full OEM Capabilities Type* == Unchecked
Default Feature Capability 1 in Hex (0x)	—	32'h00007600	Uneditable Viewable only if *Enable Full OEM Capabilities Type* == Unchecked
OEM Feature Capability 0 in Hex (0x)	—	32'h00000000	Viewable and editable only if *Enable Full OEM Capabilities Type* == Checked
OEM Feature Capability 1 in Hex (0x)	—	32'h00000000	Viewable and editable only if *Enable Full OEM Capabilities Type* == Checked
Automatically move to Configuration State	Checked, Unchecked	Unchecked	Editable only if *IP Mode* == SCM

Attribute	Selectable Values	Default	Dependency on Other Attributes
LL GPIO			
LL GPIO Input Data Width	1–96	16	Editable only if *Enable Low Latency GPIO Channel ==* Enabled
LL GPIO Output Data Width	1–96	16	Editable only if *Enable Low Latency GPIO Channel ==* Enabled
LL GPIO Payload Width per Frame[2]	8–96	16	Editable only if *Data Frame Type* == Custom and *Enable Low Latency GPIO Channel* == Enabled and *Enable Data Channel* == Disabled GPIO Data Width = whichever is greater between *GPIO Input Data Width* and *GPIO Output Data Width* *LL GPIO Payload Width per Frame* >= Integer(ceil(GPIO Data Width/8)) × 8
NL GPIO			
NL GPIO Input Data Width	1–256	16	Editable only if *Enable Normal Latency GPIO Channel* == Enabled
NL GPIO Output Data Width	1–256	16	Editable only if *Enable Normal Latency GPIO Channel* == Enabled
NL GPIO Payload Width per Frame[2]	8–96	16	Editable only if *Data Frame Type* == Custom and *Enable Normal Latency GPIO Channel* == Enabled Suggested *max* value when *Data Frame Type ==* Custom : GPIO Data Width = whichever is greater between *GPIO Input Data Width* and *GPIO Output Data Width* *max* = Integer(ceil(GPIO Data Width/8)) × 8
I^{2}C			
Number of I^{2}C bus interface	1–24	6	Editable only if *Enable I^{2}C Channel* == Enabled
I^{2}C Bus interface per Frame[2]	2–24	6	Editable only if *Data Frame Type* == *Custom* and *Enable I^{2}C Channel* == Enabled Valid values are multiples of 2 *I^{2}C Bus interface per Frame* ≥ *Number of I^{2}C bus interface* Suggested *max* value when *Data Frame Type ==* Custom : *max* = (Integer(ceil(Number of I^{2}C bus interface × 4/8)) × 8)/4
I^{2}C Bus # Mode	Controller, Target	Controller	Editable only if *Enable I^{2}C Channel* == Enabled
Enable Fast Mode for Bus #	Checked, Unchecked	Unchecked	Editable only if *Enable I^{2}C Channel* == Enabled
Customize Bus # Tsu-sda	Checked, Unchecked	Unchecked	Editable only if *Enable I^{2}C Channel* == Enabled
Bus # Tsu-sda count	1–1023	1	Editable only if *Customize Bus # Tsu-sda* == Checked
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* == Enabled
Enable Flow Control	Enabled, Disabled	Enabled	Editable only if *Enable UART Channel* == Enabled
Number of UART bus interface	1–8	2	Editable only if *Enable UART Channel* == Enabled

Attribute	Selectable Values	Default	Dependency on Other Attributes
UART Bus interface per Frame[2]	2–8	2	Editable only if *Data Frame Type* == Custom and *Enable UART Channel* == Enabled Valid values are multiples of 2 *UART Bus interface per Frame >= Number of UART bus interface* Suggested *max* value when *Data Frame Type* == Custom : *max* = (Integer(ceil(Number of UART bus interface × 4/8)) × 8)/4
OEM			
OEM Data Width	1–96	32	Editable only if *Enable OEM Channel* == Enabled
OEM Payload Width per Frame[2]	8–96	32	Editable only if *Data Frame Type* == Custom and *Enable OEM Channel* == Enabled *OEM Payload Width per Frame >= OEM Data Width* Suggested *max* value when *Data Frame Type* == Custom : *max* = Integer(ceil(*OEM Data Width*/8)) × 8
OEM Capability 0 in Hex (0x)	—	8'h0	Editable only if *Enable OEM Channel* == Enabled and *Enable Full OEM Capabilities Type* == Unchecked
OEM Capability 1 in Hex (0x)	—	8'h0	Editable only if *Enable OEM Channel* == Enabled and *Data Frame Type* == Default I/O

Notes:

1. Maximum speed for MachXO3 family.
2. For each channel enabled, corresponding allocated payload bits must be byte-aligned.

Table 2.3. Attributes Description

Attribute	Description
General	
ID Number in Hex	2-byte IP Identification Number (i.e. GUID, MFR ID, Vendor ID, Board ID, Revision)
IP Mode	Indicates if IP is meant to be used for SCM or HPM. Internal State Machine is different between two modes.
LTPI Version (Major)	Indicates the major LTPI version number.
LTPI Version (Minor)	Indicates the minor LTPI version number.
LTPI Version	Displays the concatenated major and minor LTPI version.
I/O Type	I/O type for PHY interface.
Customize CRC Polynomial	When enabled, CRC polynomial and initial value can be customized.
CRC-8 Polynomial in Hex	Specifies the CRC-8 polynomial to use. This presupposes that x^8 is already set to 1, thus, most significant bit must be excluded when setting the polynomial hex value, that is Polynomial 8'h07 translates to $x^8 + x^2 + x^1 + 1$.
CRC-8 Initial Value	Initial state of CRC-8 engine.
Enable Miscellaneous Signals	When enabled, miscellaneous signals (that is error/flag signals) are brought out as port/s.
Enable Protocol Information Signals	When enabled, signals under Protocol Information section in Table 2.1 are brought out as port/s. These signals are also available in CSR.
Enable Clock Compensation	When enabled, the IP performs frame drop or frame repeat to control the CDC FIFO underflow or overflow. This is to ensure that when such case is encountered, frame alignment won't be lost. This is suggested to be enabled for cases when there is significant difference between the SCM, HPM, and IP system clock. For more details, refer to Clocks and Reset section.
Frame Settings	
Data Frame Type	Specifies the frame type for data frames. Custom – customizable by the user. Default I/O – predefined based on DC-SCM LTPI 2.0 specifications.

Attribute	Description
Enable Low Latency GPIO Channel	Enable Low Latency GPIO external channel
Enable Normal Latency GPIO Channel	Enable Normal Latency GPIO external channel
Enable I^{2}C Channel	Enable I^{2}C external channel
Enable UART Channel	Enable UART external channel
Enable OEM Channel	Enable OEM external channel
Total Number of Payload Bytes	Total number of payload bytes in the frame, computed as the summation of payload bytes per channel, excluding Data Channel. The width of the data payload is an integer number of bytes. Denoted as *K* in some parts of the document.
Enable Data Channel	Enable Data external channel When enabled, this translates to Data Frames.
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
X32 (800 MHz)	Indicates support for 800 MHz LVDS Clock
X40 (1000 MHz)	Indicates support for 1000 MHz LVDS Clock
Target System Clock (MHz)	Indicates the target system clock frequency based on highest Speed Capability enabled
Feature Capability	
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 *OEM Feature Capability 0/1 in Hex (0x).*
Type value in Hex (0x)	Customized *Capabilities Type* value.
Capabilities Type	Indicates if Advertise frame mapping follows default, IP predefined, or user-customized capabilities mapping. Refer to Table 2.11 for details.
Default Feature Capability 0 in Hex (0x)	Indicates the default Feature Capability of the IP. This information is used in Advertise frame if *Full OEM Capabilities Type* ==Unchecked. This information is combination of multiple settings for each external channels and is used during Link-training and negotiations. Refer to Advertise Frame section for details.
Default Feature Capability 1 in Hex (0x)	Indicates the default Feature Capability of the IP. This information is used in Advertise frame if *Full OEM Capabilities Type* ==Unchecked. This information is combination of multiple settings for each external channels and is used during Link-training and negotiations. Refer to Advertise Frame section for details.
OEM Feature Capability 0 in Hex (0x)	Indicates the OEM Feature Capability of the IP. This information is used in Advertise frame if *Full OEM Capabilities Type ==C*hecked. Refer to Advertise Frame section for details.
OEM Feature Capability 1 in Hex (0x)	Indicates the OEM Feature Capability of the IP. This information is used in Advertise frame if *Full OEM Capabilities Type ==C*hecked. Refer to Advertise Frame section for details.
Automatically move to Configuration State	Specifies that when *IP Mode ==* SCM, IP automatically goes to Configuration state after completing the required frame transmission for Advertise state.

Attribute	Description
LL GPIO	
LL GPIO Input Data Width	Specifies the number of LL GPIO input pins This data width is mapped to *LL GPIO Payload Width per Frame* set in LTPI frame and is updated every frame.
LL GPIO Output Data Width	Specifies the number of LL GPIO output pins This data width is mapped to *LL GPIO Payload Width per Frame* set in LTPI frame and is updated every frame.
LL GPIO Payload Width per Frame	Specifies the number of payload bytes allocated for LL GPIO per frame. This applies to both Transmit and Receive frames.
NL GPIO	
NL GPIO Input Data Width	Specifies the number of NL GPIO input pins This data width is divided into multiple frame transfer based on *NL GPIO Payload Width per Frame* set.
NL GPIO Output Data Width	Specifies the number of GPIO output pins This data width is divided into multiple frame transfer based on *NL GPIO Payload Width per Frame* set.
NL GPIO Payload Width per Frame	Specifies the number of payload bytes allocated for NL GPIO per frame. This applies to both Transmit and Receive frames.
I^{2}C	
Number of I^{2}C bus interface	Specifies the number of I^{2}C bus interfaces in the I^{2}C channel group Each I^{2}C bus interface has dedicated 4-bit data width. This bus is mapped to *I^{2}C Bus interface per Frame* set in LTPI frame and is updated every non-Data Channel frame.
I^{2}C Bus interface per Frame	Specifies the number of payload bytes allocated for I^{2}C per frame.
I^{2}C Bus # Mode	Specifies the mode of the external I^{2}C channel connected to the IP. IP just serves as an extension/relay of the connected I^{2}C channel.
Enable Fast Mode for Bus #	Enables fast mode (400 kHz) of the I^{2}C bus.
Customize Bus # Tsu-sda	This attribute is used to customize the SDA to SCL data set-up time ($T_{SU\text{-}SDA}$) of the IP. When checked, default $T_{SU\text{-}SDA}$ is overwritten by the value specified in *Bus # Tsu-sda count.* The IP's internal default value is based on four times the minimum set-up time required by I^{2}C protocol depending on I^{2}C's clock mode, that is 1000 ns (4 × 250 ns) for Standard mode (100 kHz) and 400 ns (4 × 100 ns) for Fast mode (400 kHz). Sample computation of IP's internal default $T_{SU\text{-}SDA}$: IP System Clock = 80 MHz I^{2}C Mode (*Enable Fast Mode for Bus*== Unchecked) = 100 kHz I^{2}C Standard mode $T_{SU\text{-}SDA(MIN)}$ = 250 ns IP's internal default $T_{SU\text{-}SDA} = (4 \times T_{SU\text{-}SDA(MIN)}) / T_{sys\_clk}$ = 1000 ns / (1000/80 MHz) = ~80 cycles
Bus # Tsu-sda count	Specifies the new SDA to SCL data set-up time ($T_{SU\text{-}SDA}$) in terms of number of system clock (clk_i) cycle count. Only used when *Customize Bus # Tsu-sda* == Checked.
UART	
Baud Rate	Specifies the UART baud rate.
Enable Flow Control	Enables flow control ports of 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 Each UART bus interface has dedicated 4-bit pins for each transmit and receive direction. This bus is mapped to *UART Bus interface per Frame* set in LTPI frame and is updated every non-Data Channel frame.
UART Bus interface per Frame	Specifies the number of payload bytes allocated for UART per frame. Each UART bus has 4-bit data width allocation in the payload.

Attribute	Description
OEM	
OEM Data Width	Specifies the number of OEM pins This data width is mapped to *OEM Payload Width per Frame* set in LTPI frame and is updated every non-Data Channel frame.
OEM Payload Width per Frame	Specifies the number of payload bytes allocated for OEM per frame.
OEM Capability 0 in Hex (0x)	Indicates the default OEM channel capability.
OEM Capability 1 in Hex (0x)	Indicates the default OEM channel capability.

2.4. Register Description

DC-SCM LTPI soft IP registers are shown in Table 2.4.These are accessed through APB completer interface. When Data Channel is enabled, this is accessed starting from address 0x00000400 onwards.

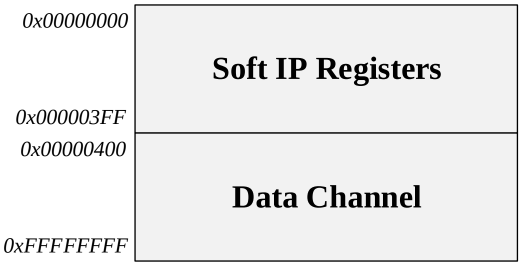


Figure 2.4. Register Address Space

Table 2.4. DC-SCM LTPI Soft IP Registers

Offset	Register Name	Access	Default Value	Register Description		
				Bit	**Field**	**Field Description**
Attribute Registers						
0x0	ID	RO	{depends on ID Number set in Attributes}	[31:16]	RSVD	Reserved bits.
				[15:0]	id	Identification Number
0x4	LTPI_VER	RO	{depends on LTPI Version (Major) and LTPI Version (Minor) set in Attributes}	[31:8]	RSVD	Reserved bits.
				[7:0]	ltpi_ver	Specifies the LTPI Version. [7:4] – Major version [3:0] – Minor version
0x8	SPEED_CAP	RO	{depends on Speed Capability set in Attributes}	[31:16]	RSVD	Reserved bits.

				[15:0]	speed_cap	Defines the speed capability of the IP. Base frequency is 25MHz.
						[7:0] = {X12,X10,X8,X6,X4,X3,X2,X1}
						[15:8] = {DDR,3'h00,X40,X32,X24,X16}

0x0C	PLATFORM_TYPE	RO	{ depends on ID Number set in Attributes}	[31:16]	RSVD	Reserved bits.
				[15:0]	platform_type	Platform type

Offset	Register Name	Access	Default Value	Register Description		
				Bit	**Field**	**Field Description**
0x10	FEATURE_CAP3_0	RO	{depends on IP and Channel Capability set in Attributes}	[31:0]	feature_cap3_0	Feature capability (byte3-0) of the IP. If *Enable Full OEM Capabilities Type* == Checked, it gets value from *OEM Feature Capability 0 in Hex (0x)*. Refer to Advertise Frame section for details.
0x14	FEATURE_CAP7_4	RO	{depends on IP and Channel Capability set in Attributes}	[31:0]	feature_cap7_4	Feature capability (byte7-4) of the IP. If *Enable Full OEM Capabilities Type* == Checked, it gets value from *OEM Feature Capability 1 in Hex (0x)*. Refer to Advertise Frame section for details.
0x18	FEATURE_CAP11_8	RO	32’h0	[31:0]	feature_cap11_8	Feature capability (byte11-8) of the IP. Currently is unused and only serves as a placeholder for future use.
0x1C	REQ_FEATURE3_0	RW	{depends on Feature Capability 0 set in Attributes}	[31:0]	req_feature3_0	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded.
0x20	REQ_FEATURE7_4	RW	{depends on Feature Capability 1 set in Attributes}	[31:0]	req_feature7_4	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded.
0x24	REQ_FEATURE 11_8	RW	32’h0	[31:0]	req_feature11_8	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded. Currently is unused and only serves as a placeholder for future use.
0x28	TGT_SPEED	RO	32’h0	[31:16]	RSVD	Reserved bits.

				[15:0]	tgt_spd	Defines the target speed of the IP based on Link training and negotiation. Base frequency is 25MHz.
						[7:0] = {X12,X10,X8,X6,X4,X3,X2,X1}
						[15:8] = {DDR,3’h00,X40,X32,X24,X16}

0x2C	RX_FEATURE3_0	RO	32’h0	[31:0]	rx_feat_cap3_0	Received Feature Capability from Remote IP during Link-training and negotiation.
0x30	RX_FEATURE7_4	RO	32’h0	[31:0]	rx_feat_cap7_4	Received Feature Capability from Remote IP during Link-training and negotiation.
0x34	RX_FEATURE 11_8	RO	32’h0	[31:0]	rx_feat_cap11_8	Received Feature Capability from Remote IP during Link-training and negotiation.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	**Field**	**Field Description**
IP Control						
0x38	IP_CTRL	RO	{23’h0, depends on Automatically move to Configuration State set in Attributes, 8’h0}	[31:9]	RSVD	Reserved bits.
				[8]	auto_cfgen	When set and IP Mode == SCM, the IP automatically goes to Configuration state after completing the required frame transmission for Advertise state.
				[7:4]	RSVD	Reserved bits.
		RW		[3]	data_ch_rst	Data channel reset register. When asserted, resets the data channel interface controller of the IP. When *IP Mode*==HPM, this also resets the APB requester interface output ports.
				[2]	software_rst	LTPI link software reset. Reset only applies to interface controller and Link-training and negotiation FSM-related logic.
				[1]	resync_link	Link retraining request register. When asserted, internally resets the IP except for IP CSR related interface. IP goes back to Link-training state. Needs to be unset to proceed with retraining after active trigger.
				[0]	reqcfg_rdy	Used to indicate access to CSR is done and requested features (for SCM) is ready for Advertise state. Bit is only checked during Advertise state. Only when this is set that SCM goes to Configuration state. When IP Mode==HPM, this register is discarded.
0x3C	I2C_BUS_RST	RO	32’h0	[31:24]	RSVD	Reserved bits.
		RW		[23:0]	i2c_bus_rst	I^{2}C bus controller reset. When asserted, IP releases the local bus and resets the interface controller of the corresponding bus link number. Each bit index is mapped to each I^{2}C bus link reset with bus link 0 occupying index 0. [0] – I^{2}C bus 0 reset [1] – I^{2}C bus 1 reset … [23] – I^{2}C bus 23 reset

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

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

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x78	INT_ENABLE	RW	32'h0	[31:11]	RSVD	Reserved bits.
				[10]	acpt_to_en	Accept state timeout – interrupt enable.
				[9]	cfg_to_en	Configuration state timeout – interrupt enable.
				[8]	ls_to_en	Link-Speed state timeout – interrupt enable.
				[7]	rx_feat_vld_en	Received feature capability is valid – interrupt enable.
				[6]	clk_cfg_en	Clock can be reconfigured for target frequency based – interrupt enable.
				[5]	tgt_spd_vld_en	Target speed information from link training and negotiation is already valid - – interrupt enable.
				[4]	link_err_en	Link error is encountered during normal operation – interrupt enable.
				[3]	rxfifo_full_en	Rx FIFO full asserted unexpectedly during normal operation – interrupt enable.
				[2]	txfifo_full_en	Tx FIFO full asserted unexpectedly during normal operation – interrupt enable.
				[1]	lol_rx_en	Loss of enable: detected deassertion of start_rx_i – interrupt enable.
				[0]	lol_tx_en	Loss of enable: detected deassertion of start_tx_i – interrupt enable.
0x7C	INT_SET	WO	32'h0	[31:11]	RSVD	Reserved bits.
				[10]	acpt_to_set	Accept state timeout
				[9]	cfg_to_set	Configuration state timeout
				[8]	ls_to_set	Link-Speed state timeout
				[7]	rx_feat_vld_set	Received feature capability is valid – set.
				[6]	clk_cfg_set	Clock can be reconfigured for target frequency based – set.
				[5]	tgt_spd_vld_set	Target speed information from link training and negotiation is already valid - – set.
				[4]	link_err_set	Link error is encountered during normal operation – set.
				[3]	rxfifo_full_set	Rx FIFO full asserted unexpectedly during normal operation – set.
				[2]	txfifo_full_set	Tx FIFO full asserted unexpectedly during normal operation – set.
				[1]	lol_rx_set	Loss of enable: detected deassertion of start_rx_i – set.
				[0]	lol_tx_set	Loss of enable: detected deassertion of start_tx_i – set.

Note:

1. Refer to the Lattice Interrupt Interface (LINTR) User Guide (FPGA-UG-02039) for details of these registers.

Table 2.5. 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

2.5. Implementation Flow

2.5.1. Tunneling

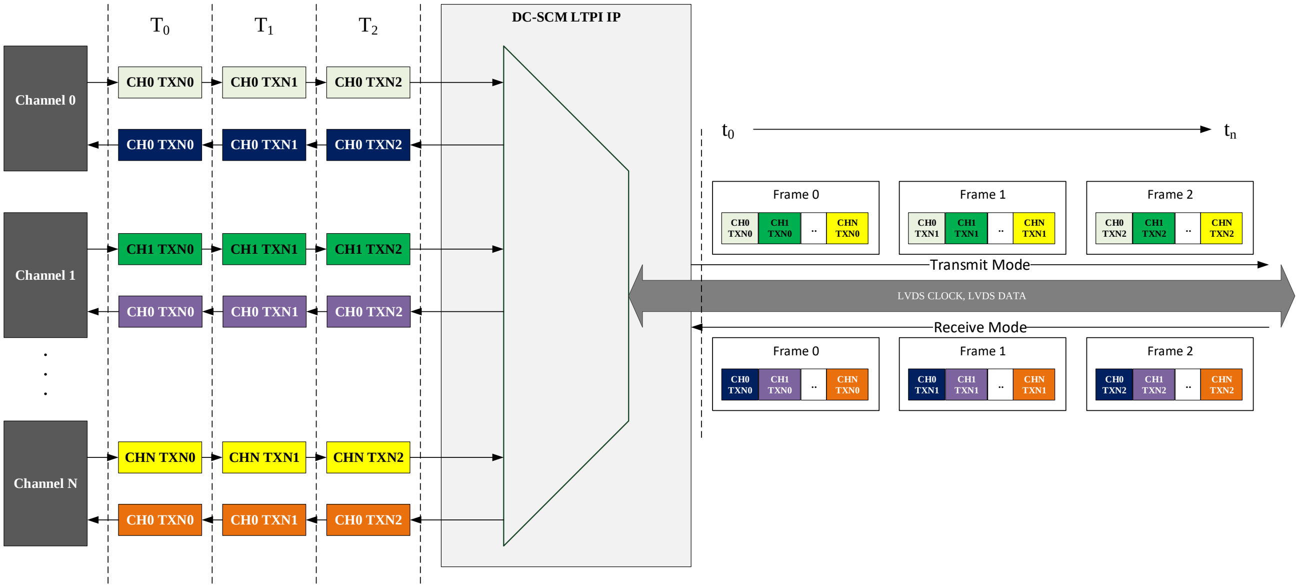
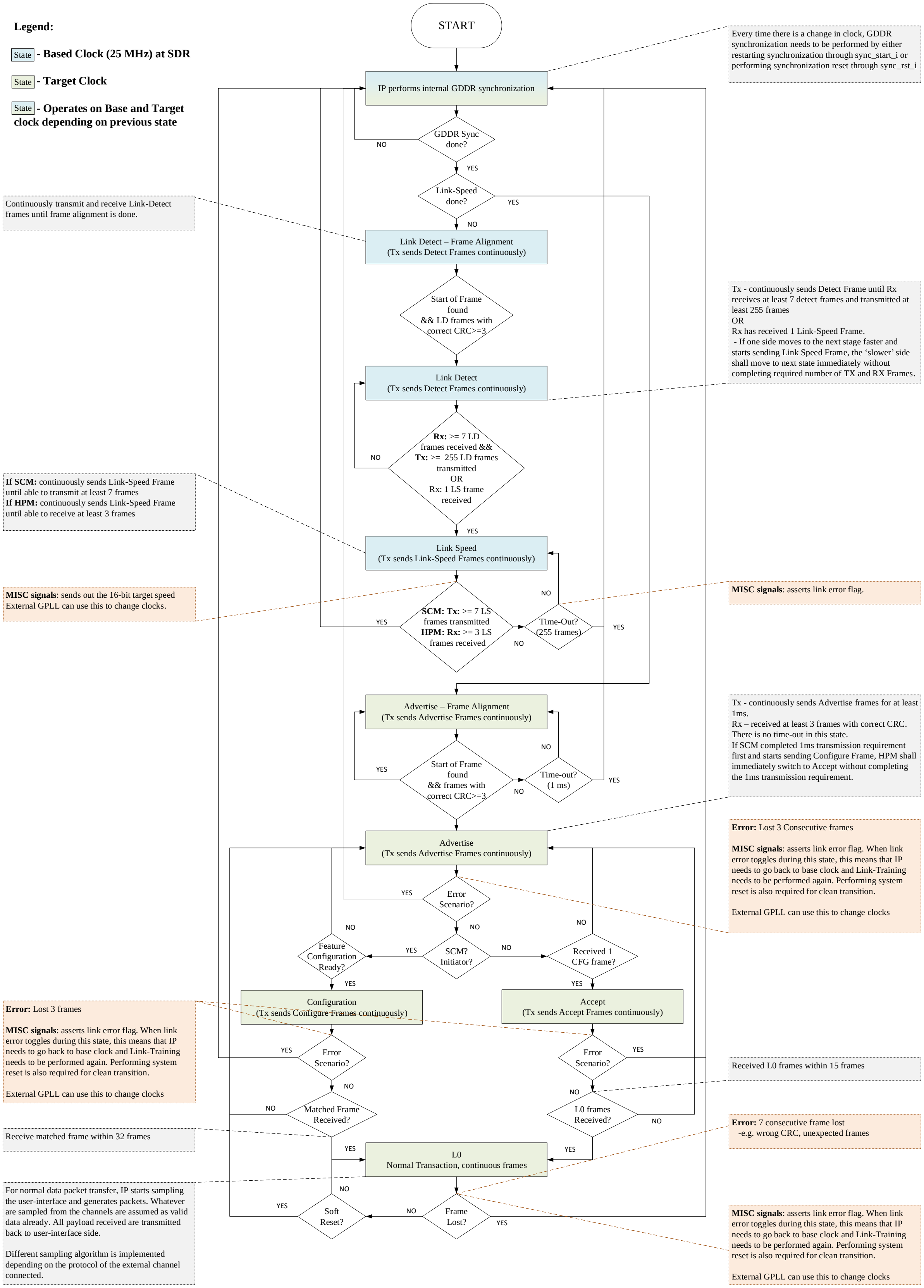


Figure 2.5. Tunneling Diagram

2.5.2. State Machine


Legend:
State - Based Clock (25 MHz) at SDR
State - Target Clock
State - Operates on Base and Target clock depending on previous state
START
IP performs internal GDDR synchronization
GDDR Sync done?
NO
YES
Link-Speed done?
YES
NO
Link Detect – Frame Alignment (Tx sends Detect Frames continuously)
Start of Frame found && LD frames with correct CRC>=3
Link Detect (Tx sends Detect Frames continuously)
NO
Rx: >= 7 LD frames received && Tx: >= 255 LD frames transmitted OR Rx: 1 LS frame received
YES
Link Speed (Tx sends Link-Speed Frames continuously)
NO
YES
SCM: Tx: >= 7 LS frames transmitted HPM: Rx: >= 3 LS frames received
NO
Time-Out? (255 frames)
YES
Advertise – Frame Alignment (Tx sends Advertise Frames continuously)
NO
YES
Start of Frame found && frames with correct CRC>=3
NO
Time-out? (1 ms)
YES
Advertise (Tx sends Advertise Frames continuously)
YES
Error Scenario?
NO
NO
NO
Feature Configuration Ready?
YES
SCM? Initiator?
NO
Received 1 CFG frame?
YES
YES
Configuration (Tx sends Configure Frames continuously)
Accept (Tx sends Accept Frames continuously)
YES
Error Scenario?
Error Scenario?
YES
NO
NO
NO
Matched Frame Received?
L0 frames Received?
NO
YES
YES
L0 Normal Transaction, continuous frames
NO
YES
Soft Reset?
NO
Frame Lost?
YES

Every time there is a change in clock, GDDR synchronization needs to be performed by either restarting synchronization through sync_start_i or performing synchronization reset through sync_rst_i

Continuously transmit and receive Link-Detect frames until frame alignment is done.

Tx - continuously sends Detect Frame until Rx receives at least 7 detect frames and transmitted at least 255 frames
OR
Rx has received 1 Link-Speed Frame.
- If one side moves to the next stage faster and starts sending Link Speed Frame, the 'slower' side shall move to next state immediately without completing required number of TX and RX Frames.

If SCM: continuously sends Link-Speed Frame until able to transmit at least 7 frames
If HPM: continuously sends Link-Speed Frame until able to receive at least 3 frames

MISC signals: sends out the 16-bit target speed External GPLL can use this to change clocks.

MISC signals: asserts link error flag.

Tx - continuously sends Advertise frames for at least 1ms.
Rx – received at least 3 frames with correct CRC.
There is no time-out in this state.
If SCM completed 1ms transmission requirement first and starts sending Configure Frame, HPM shall immediately switch to Accept without completing the 1ms transmission requirement.

Error: Lost 3 Consecutive frames

MISC signals: asserts link error flag. When link error toggles during this state, this means that IP needs to go back to base clock and Link-Training needs to be performed again. Performing system reset is also required for clean transition.

External GPLL can use this to change clocks

Error: Lost 3 frames

MISC signals: asserts link error flag. When link error toggles during this state, this means that IP needs to go back to base clock and Link-Training needs to be performed again. Performing system reset is also required for clean transition.

External GPLL can use this to change clocks

Received L0 frames within 15 frames

Receive matched frame within 32 frames

Error: 7 consecutive frame lost
-e.g. wrong CRC, unexpected frames

For normal data packet transfer, IP starts sampling the user-interface and generates packets. Whatever are sampled from the channels are assumed as valid data already. All payload received are transmitted back to user-interface side.

Different sampling algorithm is implemented depending on the protocol of the external channel connected.

MISC signals: asserts link error flag. When link error toggles during this state, this means that IP needs to go back to base clock and Link-Training needs to be performed again. Performing system reset is also required for clean transition.

External GPLL can use this to change clocks

Figure 2.6. FSM of DC-SCM LTPI IP

2.5.2.1. PHY states

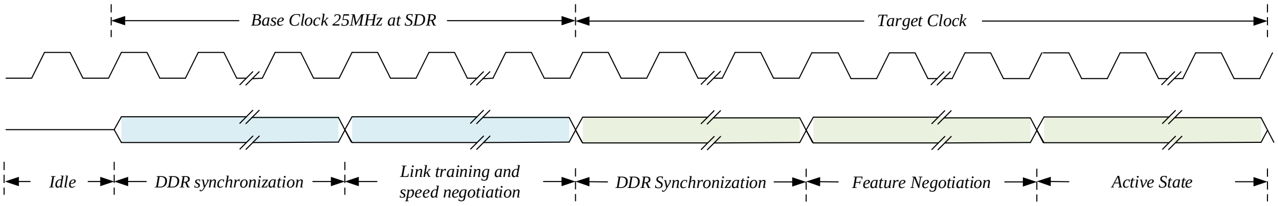


Figure 2.7. PHY States

2.5.2.2. GDDR Clock Synchronization State

During initialization, IP must commence start-up by performing internal DDR synchronization through GDDR synchronization control ports. GDDR synchronization is required by the DDR interface to properly synchronize DDR related clocks. Successful GDDR synchronization is indicated by the assertion of sync_ready_o.

Every time there is a change in working frequency of the IP, GDDR synchronization must be performed by either restarting synchronization through sync_start_i or performing synchronization reset through sync_rst_i. Synchronization must be done only after target clock becomes stable. GDDR clock synchronization logic is always clocked by sync_clk_i.

2.5.2.3. Link Training – Link-Detect State

Link-Detect state is divided into two subparts:

1. Link-Detect – Frame Alignment: After initial DDR clock synchronization, 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 2.7. Tx continuously transmits Link-Detect Frame while Rx continuously receives the frames. 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, state transitions to Link-Detect after three frames with correct CRC are received.
2. Link-Detect: Frame counter is started after frame alignment is done. When Rx has already receive at least seven consecutive Link-Detect Frames and Tx has transmitted at least 255 Link-Detect Frames, IP transitions to 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 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. 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. Content of Link-Speed Frame is shown in Table 2.8.

To transition to Advertise state from Link-Speed state, following conditions must be satisfied:

- If SCM: Tx has already transmitted at least 7 Link-Speed Frames
- If HPM: Rx has already received at least 3 Link-Speed frame with the same target speed set

If after 255 Link-Speed Frames are received and no same target speed is detected in the received frames, IP goes back to Link-Detect state and link_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:

1. 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 2.9. Advertise frame is continuously transmitted between SCM and HPM for at least 1 ms. 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.

2. Advertise – This is the main part of Advertise where Frames are being used to interpret the LTPI capabilities of the other side.

Both SCM and HPM shall keep sending Advertise Frames for at least 1ms to allow the link to stabilize at the operational frequency. If three consecutive frames are lost during Advertise state, IP goes back to GDDR synchronization state and link_err_o port is asserted. Initialization and Link-Training must be performed again.

If at least three consecutive Advertise Frames are received and Tx has already transmitted Advertise frames for at least 1 ms, depending on the *IP Mode* set in the attribute, IP transitions to different states. If IP is set as SCM, 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 in order to move the IP to Configuration state. If IP is set as HPM, IP goes to *Accept* state when it receives at least one Configuration Frame from SCM. If SCM completed 1ms transmission requirement first and already starts sending Configure Frame, HPM shall immediately switch to Accept without completing the 1ms transmission requirement.

In cases when the IP goes back to Advertise state from either Configuration or Accept state due to feature mismatch, user must set Configuration ready again.

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 is shown in Table 2.16. If a matched Accept Frame from HPM is received, IP goes to L0 state. If within 32 frames and matched Accept Frame is not received, IP goes back to Advertise state and IP_CTRL.reqcfg_rdy is automatically cleared. A matched frame is equivalent to Accept Frame from HPM with the same enabled features as Configure Frame sent by SCM (request feature bits equals to accept feature bits). If three frames are lost, the IP goes back to GDDR synchronization state and link_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 Frames is the AND logic of feature capability (Advertise Frame) and requested features (Configure Frame) and frame content is shown in Table 2.17, with the exception of UART feature field which is based on highest baud rate supported. IP continuously transmits Accept Frames to SCM until it receives L0 Frames. If L0 Frames are not received within 15 frames, IP goes back to Advertise state. If three frames are lost, the IP goes back to GDDR synchronization state and link_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. If three consecutive frames are lost or system reset is detected, IP goes back to DDR synchronization state and initialization and Link-Training must be performed again. If frame lost is detected, link_err_o port is asserted. If soft reset is detected, IP goes back to Advertise state.

2.6. Frame Format

2.6.1. Frame Format

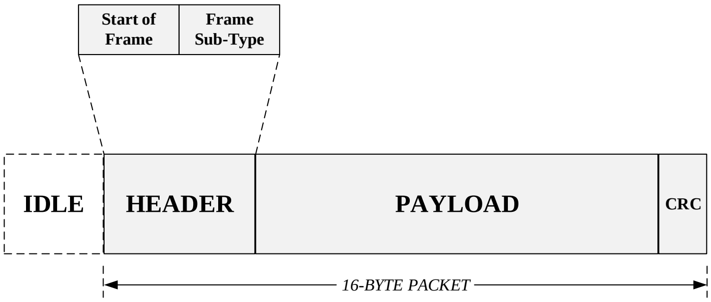


Figure 2.8 Frame Illustration

Table 2.6. Frame Format Contents

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

Table 2.6 illustrates the frame format. First byte indicates the start of frame and this depends on the frame format (that is Link-Training frames, Data frames). This is followed by an 8-bit symbol that indicates the frame sub-type, the 13-byte frame data and the 8-bit CRC byte of the frame.

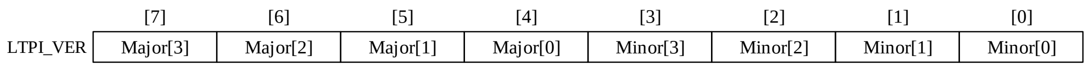
2.6.2. Different Frame Formats

2.6.2.1. Link-Detect Frame

Link-Detect Frames are transmitted during Link Training – Link-Detect state. Base frequency for Speed Capability is 25 MHz.

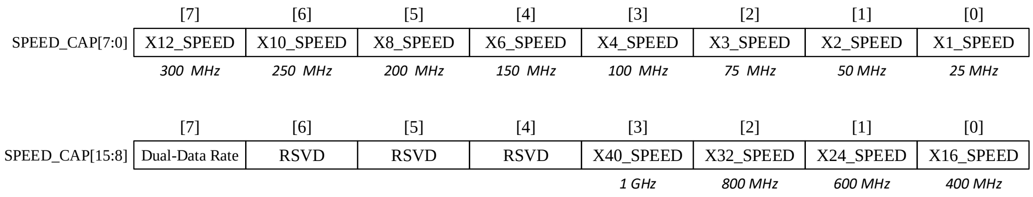
Table 2.7 Link-Detect Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.5	LT_SYMBOL
1	D0.0	LDFT_ SYMBOL
2	LTPI Version	LTPI_VER
3	Speed Capability 1	SPEED_CAP[7:0]
4	Speed Capability 2	SPEED_CAP[15:8]
5	Reserved	IDLE_SYMBOL
6	Reserved	IDLE_SYMBOL
7	Reserved	IDLE_SYMBOL
8	Reserved	IDLE_SYMBOL
9	Reserved	IDLE_SYMBOL
10	Reserved	IDLE_SYMBOL
11	Reserved	IDLE_SYMBOL
12	Reserved	IDLE_SYMBOL
13	Reserved	IDLE_SYMBOL
14	Reserved	IDLE_SYMBOL
15	CRC	N/A (Computed by the IP)



	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
LTPI_VER	Major[3]	Major[2]	Major[1]	Major[0]	Minor[3]	Minor[2]	Minor[1]	Minor[0]

Figure 2.9. LTPI_VER Byte Mapping



	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
SPEED_CAP[7:0]	X12_SPEED	X10_SPEED	X8_SPEED	X6_SPEED	X4_SPEED	X3_SPEED	X2_SPEED	X1_SPEED
	300 MHz	*250 MHz*	*200 MHz*	*150 MHz*	*100 MHz*	*75 MHz*	*50 MHz*	*25 MHz*

	[7]	[6]	[5]	[4]	[3]	[2]	[1]	[0]
SPEED_CAP[15:8]	Dual-Data Rate	RSVD	RSVD	RSVD	X40_SPEED	X32_SPEED	X24_SPEED	X16_SPEED
					1 GHz	*800 MHz*	*600 MHz*	*400 MHz*

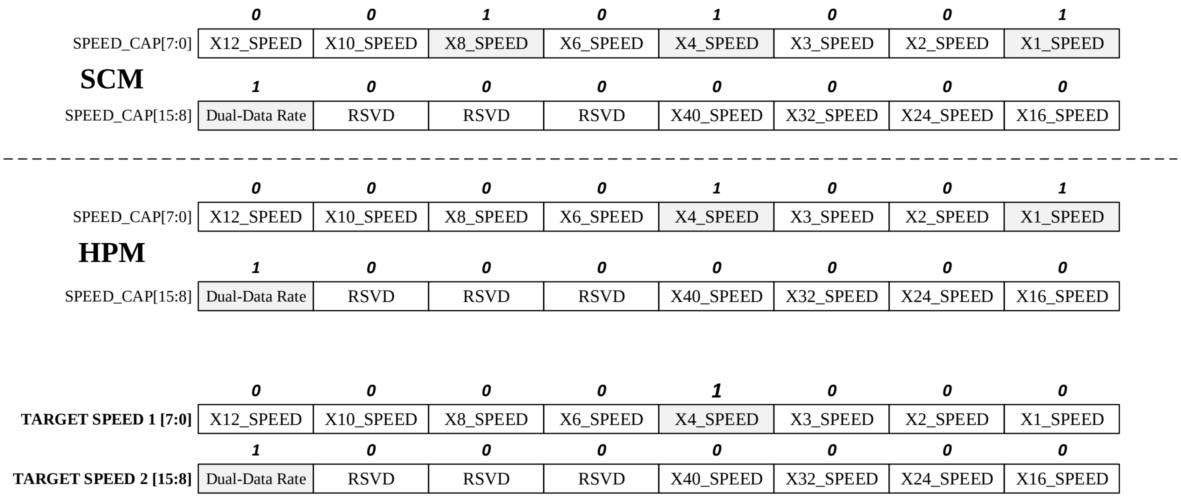
Figure 2.10. SPEED_CAP Byte Mapping

2.6.2.2. Link-Speed Frame

Link-Speed Frames are transmitted during Link Training – Link-Speed state.

Table 2.8. Link-Speed Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.5	LT_SYMBOL
1	D1.0	LSFT_ SYMBOL
2	LTPI Version	LTPI_VER
3	Target Speed 1	N/A Processed by the IP
4	Target Speed 2	
5	Reserved	IDLE_SYMBOL
6	Reserved	IDLE_SYMBOL
7	Reserved	IDLE_SYMBOL
8	Reserved	IDLE_SYMBOL
9	Reserved	IDLE_SYMBOL
10	Reserved	IDLE_SYMBOL
11	Reserved	IDLE_SYMBOL
12	Reserved	IDLE_SYMBOL
13	Reserved	IDLE_SYMBOL
14	Reserved	IDLE_SYMBOL
15	CRC	N/A (Computed by the IP)



SCM

	0	*0*	*1*	*0*	*1*	*0*	*0*	*1*
SPEED_CAP[7:0]	X12_SPEED	X10_SPEED	X8_SPEED	X6_SPEED	X4_SPEED	X3_SPEED	X2_SPEED	X1_SPEED
	1	*0*	*0*	*0*	*0*	*0*	*0*	*0*
SPEED_CAP[15:8]	Dual-Data Rate	RSVD	RSVD	RSVD	X40_SPEED	X32_SPEED	X24_SPEED	X16_SPEED

HPM

	0	*0*	*0*	*0*	*1*	*0*	*0*	*1*
SPEED_CAP[7:0]	X12_SPEED	X10_SPEED	X8_SPEED	X6_SPEED	X4_SPEED	X3_SPEED	X2_SPEED	X1_SPEED
	1	*0*	*0*	*0*	*0*	*0*	*0*	*0*
SPEED_CAP[15:8]	Dual-Data Rate	RSVD	RSVD	RSVD	X40_SPEED	X32_SPEED	X24_SPEED	X16_SPEED

	0	*0*	*0*	*0*	*1*	*0*	*0*	*0*
TARGET SPEED 1 [7:0]	X12_SPEED	X10_SPEED	X8_SPEED	X6_SPEED	X4_SPEED	X3_SPEED	X2_SPEED	X1_SPEED
	1	*0*	*0*	*0*	*0*	*0*	*0*	*0*
TARGET SPEED 2 [15:8]	Dual-Data Rate	RSVD	RSVD	RSVD	X40_SPEED	X32_SPEED	X24_SPEED	X16_SPEED

Figure 2.11. Target Speed Processing Illustration

Target speed is determined based on the fastest clock capability that is common between SCM and HPM. In the example in Figure 2.11, SCM 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.

2.6.2.3. Advertise Frame

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

Table 2.9. Advertise Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	CFG_SYMBOL
1	D0.0	ADVFT_SYMBOL
2	Platform Type Byte 0	PLATFORM_TYPE[7: 0]
3	Platform Type Byte 1	PLATFORM_TYPE[15: 8]
4	Capabilities Type	N/A
5	Feature Capability 1	FEATURE_CAP3_0[7: 0]
6	Feature Capability 2	FEATURE_CAP3_0[15: 8]
7	Feature Capability 3	FEATURE_CAP3_0[23:16]
8	Feature Capability 4	FEATURE_CAP3_0[31:24]
9	Feature Capability 5	FEATURE_CAP7_4[7: 0]
10	Feature Capability 6	FEATURE_CAP7_4[15: 8]
11	Feature Capability 7	FEATURE_CAP7_4[23:16]
12	Feature Capability 8	FEATURE_CAP7_4[31:24]
13	Reserved	N/A
14	Reserved	N/A
15	CRC	N/A (Computed by the IP)

Platform Field bit mapping is shown in Table 2.10.

Table 2.10. Platform Field Bit Mapping

<table>
<tr><th rowspan="2">Byte Sequence</th><th colspan="8">Bit Field</th></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="8" rowspan="2">ID (16-bit ID Number set in Attributes)</td></tr>
<tr><td>1</td></tr>
</table>

Capabilities Type value is shown in Table 2.11.

Table 2.11. Capabilities Type Details

Capabilities Type	Description
0x00	Used for default Feature Capability mapping defined in LTPI Specifications as shown in Table 2.12.
0x01-0x80	LTPI Reserved
0x81	Used for predefined Feature Capability mapping defined by the IP for custom type as shown in Table 2.13.
0x82–0xFF	OEM defined When defined, Feature Capability mapping is as shown in Table 2.14.

Feature Capability mapping is dependent on the Capabilities Type Selected. When OEM defined is used, the IP uses the default feature settings set during IP generation for each external channels (for example, I^{2}C bus follows the speed bus 100 kHz/400 kHz set during IP generation). Definition of actual feature mapping for OEM defined capability type is out of scope of the soft IP.

Table 2.12. Feature Capability Mapping for Default I/O Frame (0x00)

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	External Channel Enable							
	RSVD			OEM	Data Channel	UART	I^{2}C	GPIO
1	Total Number of NL GPIO[7:0]1							
2	RSVD						Total Number of NL GPIO[9:8]	
3	RSVD	Echo support[2] 0:Disabled 1: Enabled	I^{2}C BUS5[3] 0: Disabled 1: Enabled	I^{2}C BUS4[3] 0: Disabled 1: Enabled	I^{2}C BUS3[3] 0: Disabled 1: Enabled	I^{2}C BUS2[3] 0: Disabled 1: Enabled	I^{2}C BUS1[3] 0: Disabled 1: Enabled	I^{2}C BUS0[3] 0: Disabled 1: Enabled
4	I^{2}C Channel Capabilities							
	RSVD		I^{2}C BUS5 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS4 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS3 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS2 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS1 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS0 SP 0: 100 kHz 1: 400 kHz
5	UART Channel Capabilities							
	RSVD	UART1 0: Disabled 1: Enabled	UART0 0: Disabled 1: Enabled	Flow Control	Baud Rate Encoding (refer to Table 2.15)			
6	OEM Capability 0							
7	OEM Capability 1							

Notes:

1. Total Number of NL GPIO is the summation of input and output NL GPIO set in the IP.
2. Echo support is always enabled for I^{2}C 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.

Table 2.13. Feature Capability Mapping for Custom I/O Frame (0x81)

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
0	External Channel Enable							
	RSVD		OEM	Data Channel	UART	I^{2}C	NL GPIO	LL GPIO
1	Total Number of NL GPIO[7:0][1]							
2	RSVD						Total Number of NL GPIO[9:8]	
3	UART Channel Capabilities							
	RSVD			Flow Control	Baud Rate Encoding (refer to Table 2.15)			
4	I^{2}C Channel Capabilities[2]							
	I^{2}C BUS7 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS6 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS5 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS4 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS3 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS2 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS1 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS0 SP 0: 100 kHz 1: 400 kHz
5	Additional I^{2}C Channel Capabilities							
	I^{2}C BUS15 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS14 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS13 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS12 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS11 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS10 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS9 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS8 SP 0: 100 kHz 1: 400 kHz

Byte Sequence	Bit Field							
	7	6	5	4	3	2	1	0
6	Additional I^{2}C Channel Capabilities							
	I^{2}C BUS23 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS22 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS21 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS20 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS19 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS18 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS17 SP 0: 100 kHz 1: 400 kHz	I^{2}C BUS16 SP 0: 100 kHz 1: 400 kHz
7	OEM Capability 0							

Notes:

1. Total Number of NL GPIO is the summation of input and output NL GPIO set in the IP.
2. Echo support is always enabled for I^{2}C channel.

Table 2.14. Feature Capability Mapping for OEM Defined (0x82-0xFF)

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

Table 2.15. 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 is consists of the requested features set by SCM.

Table 2.16. Configure Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	CFG_SYMBOL
1	D1.0	REQFT_SYMBOL
2	Capabilities Type	N/A
3	Request Feature 1	REQ_FEATURE3_0[7:0]
4	Request Feature 2	REQ_FEATURE3_0[15:8]

Byte Sequence	Symbol	Corresponding Register
5	Request Feature 3	REQ_FEATURE3_0[23:16]
6	Request Feature 4	REQ_FEATURE3_0[31:24]
7	Request Feature 5	REQ_FEATURE7_4[7:0]
8	Request Feature 6	REQ_FEATURE7_4[15:8]
9	Request Feature 7	REQ_FEATURE7_4[23:16]
10	Request Feature 8	REQ_FEATURE7_4[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)

2.6.2.5. Accept Frame

Accept Frames are transmitted by HPM during Accept state. This frame indicates which features are accepted by the HPM.

Table 2.17. Accept Frame Format

Byte Sequence	Symbol	Corresponding Register
0	K28.6	CFG_SYMBOL
1	D2.0	ACCFT_SYMBOL
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)

Accept Frame is the AND logic of feature capability (Advertise Frame) of HPM and requested features (Configure Frame) from SCM with the exception of UART feature field which is based on highest baud rate supported. For UART field, requested baud rate is accepted as long as 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 requested features (Configure Frame) from SCM.

2.6.2.6. Data/I/O Frames

Data/I/O frames are transmitted during L0 state. Depending on which *Data Frame Type* is selected, frame is mapped accordingly. Data frame payload mapping and frame sub-type summary is shown in Table 2.18.

Table 2.18. 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
I^{2}C 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 fields are customizable by the user. Tx generates a 12-byte wide payload with length varying depending on the payload size set per frame per channel. If user-set payload length is less than 12-bytes, payload is padded with 0s to complete the 12-byte payload. NL Frame Counter always occupies 15th byte of Custom I/O frame.

Least significant byte of payload is automatically allocated to channel 0. Order of payload allocation is as follows: Low Latency GPIO, Normal Latency GPIO, I^{2}C, UART, OEM, with Low Latency GPIO set to occupy the least significant byte of the payload (Payload_byte[0]). If any of the channels is disabled, the next priority channel takes its payload field. Depending on number of payload bits set per channel, allocation follows an incrementing order for channel and byte assignment.

Table 2.19 lists a sample Custom I/O frame with user-custom payload setting of 2 bytes LL GPIO, 2 bytes NL GPIO, 6 I^{2}C buses, 2 UART buses, and 4 bytes OEM.

Table 2.19. Sample Custom I/O Format

Byte Sequence	Symbol	Corresponding Register
0	K28.7	SP_SYMBOL
1	0x10	DFT_SYMBOL
2	LL GPIO 01	N/A Processed by the IP
3	LL GPIO 02	
4	NL GPIO 01	
5	NL GPIO 02	
6	I^{2}C Bus 1 and 0	
7	I^{2}C Bus 3 and 2	
8	I^{2}C 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 is predefined based on DC-SCM LTPI standard. This is used to aggregate normal and low latency GPIOs, I^{2}C, UART, and OEM with fixed number of channels and frame allocation bytes. NL Frame Counter always occupies third byte of Default I/O frame.

Table 2.20. 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	I^{2}C Bus 1 and 0	
9	I^{2}C Bus 3 and 2	
10	I^{2}C 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* == Disabled, LL GPIO 01/02 are set to high by default.

Table 2.21. 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 Comma Symbol (Frame Sub-Type plus payload).

Depending on the polynomial set in the *CRC-8 Polynomial in Hex* attribute, the IP computes for the payload CRC-8. Default CRC-8 definition is shown below.

- Polynomial – $x^8 + x^2 + x^1 + 1$
- CRC-8 Polynomial in Hex – 8'h07
- CRC-8 Initial Value – 8'h00

2.6.4. Frame Interleave

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

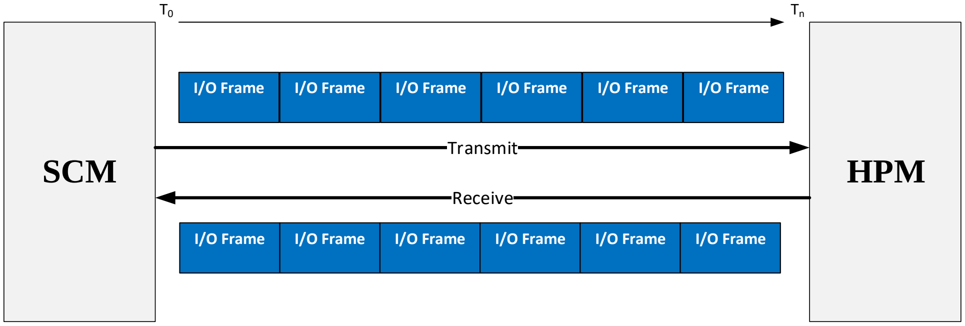


Figure 2.12. LTPI Frame Stream without Data Channel

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.

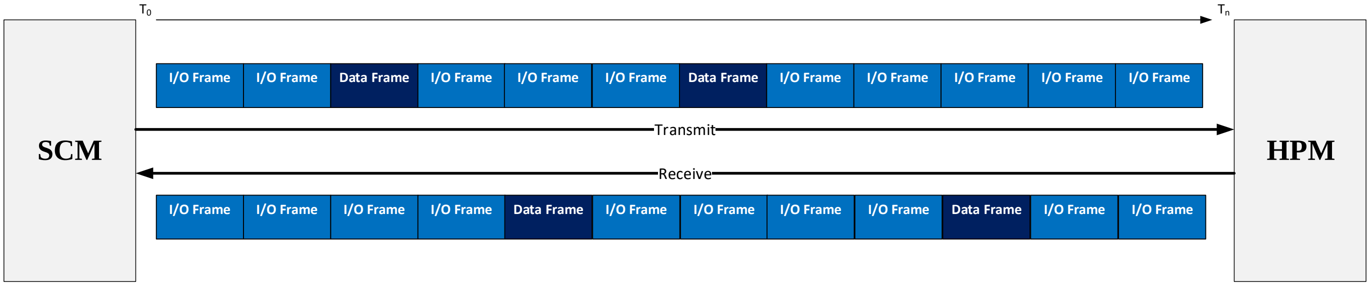


Figure 2.13. LTPI Frame Interleave with Data Channel

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 2.14 shows the sample waveform. Assuming each channel has 1 byte allocation in the frame payload, and synchronous sampling of payload happened at positive edge of t_1, resulting frame is shown in Figure 2.15.

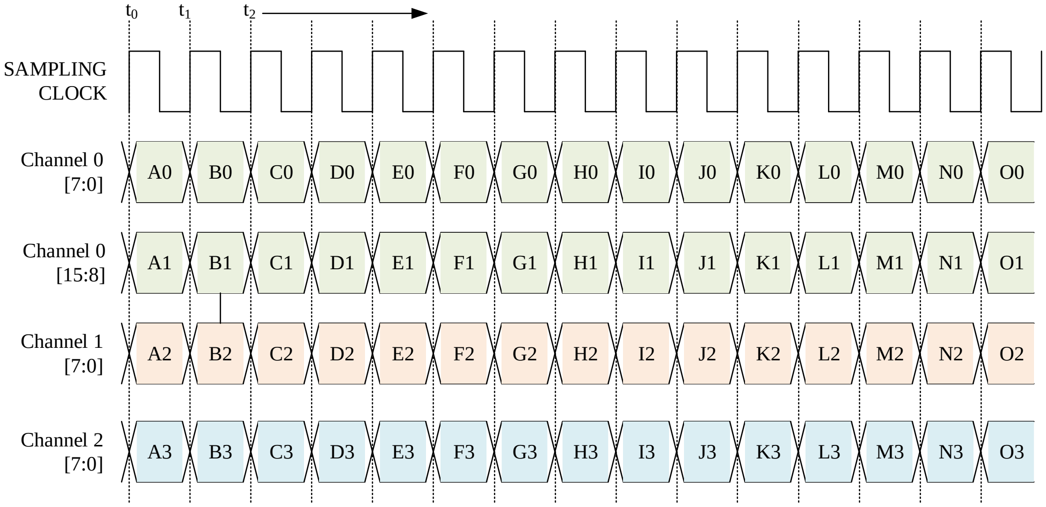

t0
t1
t2
SAMPLING CLOCK
Channel 0 [7:0]
A0 B0 C0 D0 E0 F0 G0 H0 I0 J0 K0 L0 M0 N0 O0
Channel 0 [15:8]
A1 B1 C1 D1 E1 F1 G1 H1 I1 J1 K1 L1 M1 N1 O1
Channel 1 [7:0]
A2 B2 C2 D2 E2 F2 G2 H2 I2 J2 K2 L2 M2 N2 O2
Channel 2 [7:0]
A3 B3 C3 D3 E3 F3 G3 H3 I3 J3 K3 L3 M3 N3 O3

Figure 2.14. Sample waveform

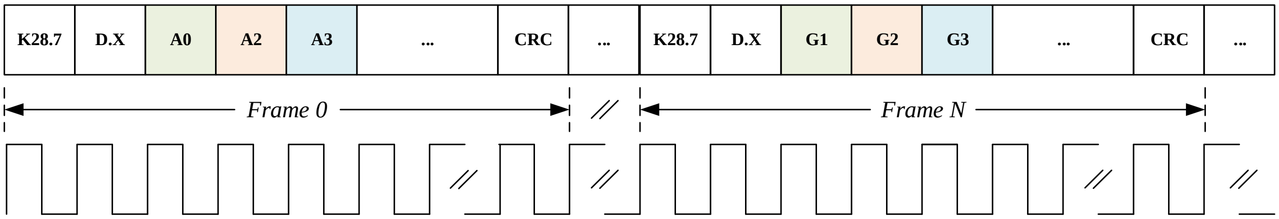


Figure 2.15. Resulting Frame

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 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 2.8. One complete frame is consists of header, payload, and footer. Actual content of the frame depends on the type of frame being generated/parsed. Refer to 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 receiving host. For Tx, 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 Rx, the 10-bit data received from 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. IP serializes the data through generic DDR interface. Likewise for the Rx mode, data is de-serialized through 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. Least significant bit of the parallel data is transmitted first in LVDS data bus.

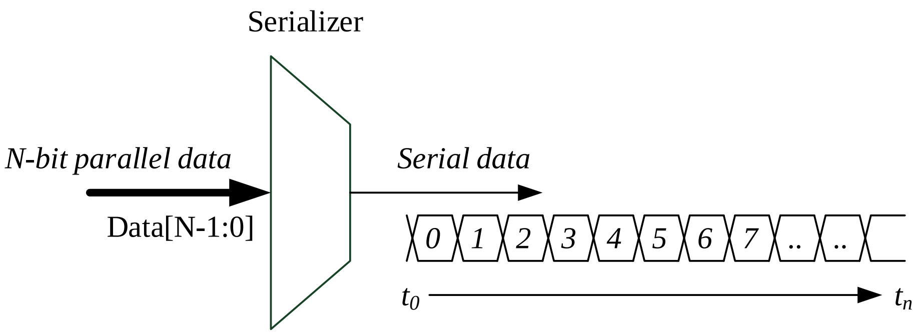


Figure 2.16. Parallel to Serial Conversion Order

2.7.5. Word Aligner

Word aligner is used by GDDR Rx to align the incoming data. After GDDR synchronization, word aligner continuously rotate the received parallel data by asserting ALIGNWD port of GDDR Rx module until it is able to get the correct word alignment. This ALIGNWD port is used by GDDR to rotate the data by 1 bit.

Depending on the current state of the IP, different patterns are used by the word aligner. During Link-Training, *LT_SYMBOL* and *LDFT_SYMBOL* are used. During Advertise state, *CFG_SYMBOL* and *ADVFT_SYMBOL* are used.

2.7.6. CSR

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 complete list of supported registers, refer to Table 2.4.

2.8. IP Programming Sequence

Clock switching and feature capability information are available in both CSR and IP ports. User can opt to get information from either IP ports when *Enable Protocol Information Signals == Enabled* or CSR when interrupts are enabled.

2.8.1. Clock Switching

During link-training and negotiation, SCM and HPM transmit their corresponding speed capability and IP parses this information to generate the target speed. Target speed information is available in both CSR and IP ports.

If interrupt is enabled, once IP is able to negotiate the target speed, the INT_STATUS.tgt_spd_vld_int interrupt is asserted to indicate that target speed information is available. Target speed information is available in register TGT_SPD. After reading this register, INT_STATUS must be cleared. Similarly, such information are also available in IP ports tgt_spd_o and tgt_spd_vld_o. When tgt_spd_vld_o is asserted, it indicates that target speed is already valid and available in tgt_spd_o.

If interrupt is enabled, when IP is ready for clock reconfiguration, the INT_STATUS.clk_cfg_int interrupt is asserted. During this time, clock should be reconfigured to the target frequency based on the information read from TGT_SPD. Once external clock reconfiguration is done, INT_STATUS must be cleared. Likewise, such information is also available in IP port clkcfg_en_o. When asserted, it indicates clock can be reconfigured to target frequency.

2.8.2. Request Features

The feature request is primarily for *IP Mode == SCM*. Once IP is in Advertise state and feature capability is properly received, when interrupt is enabled, INT_STATUS.rx_feat_vld_int interrupt is asserted to indicate that received feature capability is already valid. Received feature capability information are available in registers RX_FEATURE3_0 and RX_FEATURE7_4. After reading these registers, INT_STATUS must be cleared. The user can use the information to determine the features to be requested. Likewise, such information are also available in IP ports rx_feat_cap_o and rx_feat_cap_vld_o. When rx_feat_cap_vld_o is asserted, it indicates that received feature capability is already valid and available in rx_feat_cap_o.

By default, REQ_FEATURE3_0 and REQ_FEATURE7_4 are set to whatever feature capability are set during IP generation. If user intends to update the feature request, these registers must be programmed accordingly.

Once request feature related programming is done, user must set IP_CTRL.reqcfg_rdy. This register is used by the IP to determine if request related feature programming is done and feature negotiation can be resumed.

If *Automatically move to Configuration State* (IP_CTRL.auto_cfgen) 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 IP_CTRL.reqcfg_rdy to be set and done.

Only when either IP_CTRL.reqcfg_rdy or IP_CTRL.auto_cfgen is properly set that IP goes to Configuration state. IP_CTRL.reqcfg_rdy is automatically set to default value when soft reset is applied during normal operation state.

2.9. External Channel Interface Handling

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

Table 2.22. 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 Frame counter that 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 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 2.23. 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 Payload Width per Frame	16
Enable Normal Latency GPIO Channel	Enabled
NL GPIO Input Data Width	64
NL GPIO Output Data Width	16
NL GPIO Payload Width per Frame	8
Enable I²C Channel	Enabled
Number of I²C bus interface	6
I²C Bus interface per Frame	6
Enable UART Channel	Enabled
Number of UART bus interface	2
UART Bus interface per Frame	2
Enable OEM Channel	Enabled

Attribute	Selected Values
OEM Data Width	16
OEM Payload Width per Frame	16
Enable Data Channel	Disabled

2.9.1. GPIO Channel

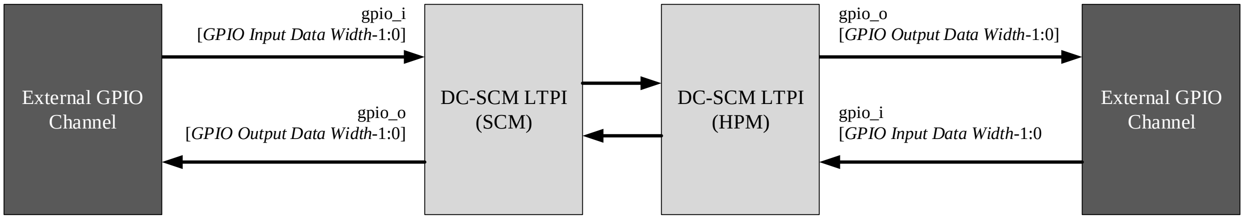

External GPIO Channel
gpio_i
[GPIO Input Data Width-1:0]
gpio_o
[GPIO Output Data Width-1:0]
DC-SCM LTPI (SCM)
DC-SCM LTPI (HPM)
gpio_o
[GPIO Output Data Width-1:0]
gpio_i
[GPIO Input Data Width-1:0
External GPIO Channel

Figure 2.17. GPIO Pin List

For normal and low latency GPIO, payload allocation for both Rx and Tx frames depends on the *NL/LL GPIO Payload Width per Frame* 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 2.23, two bytes are allocated for LL GPIO payload per frame. If there is a total of 8 *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 Tx, 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* set. 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. 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 PGIO is disabled, frame counter is fixed to 1. NL Frame Counter takes non-zero value that wraps around every *M* count based on equation below:

$$\text{If } NL\ GPIO\ Data\ Width \geq NL\ GPIO\ Payload\ Width\ per\ Frame:$$

$$M = Ceiling\left(\frac{NL\ GPIO\ Data\ Width}{NL\ GPIO\ Payload\ Width\ per\ Frame}\right)$$

Else:

$$M = 1\,;$$

In the example in Table 2.23, one byte is allocated for NL GPIO payload per frame. If there is a total of 64 *NL GPIO Input Data Width*, IP switches sampling between each 8-bit GPIO input pins per sampling time with NL Frame Counter starting at 1, until it is able to wrap around the total input pins with NL Frame Counter wrapping around at 8. For the received frames, IP remaps the received NL GPIO payload into the GPIO output pins. Depending on *NL GPIO Output Data Width,* IP switches distribution between each 8-bit GPIO output pins until it is able to wrap around the total pins. Refer to Figure 2.18 for illustration.

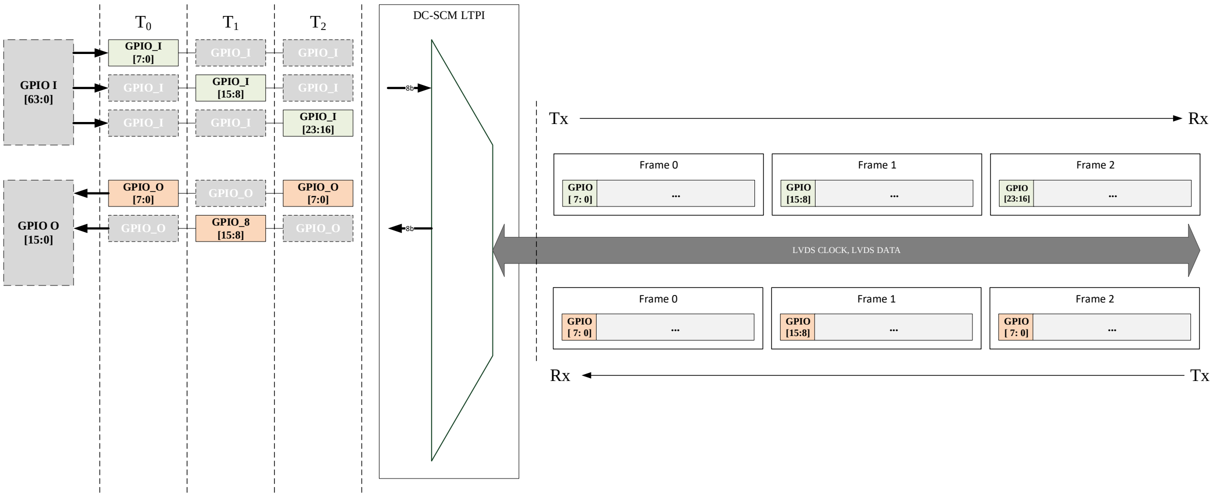


Figure 2.18. GPIO Mechanism

2.9.2. I^{2}C Channel

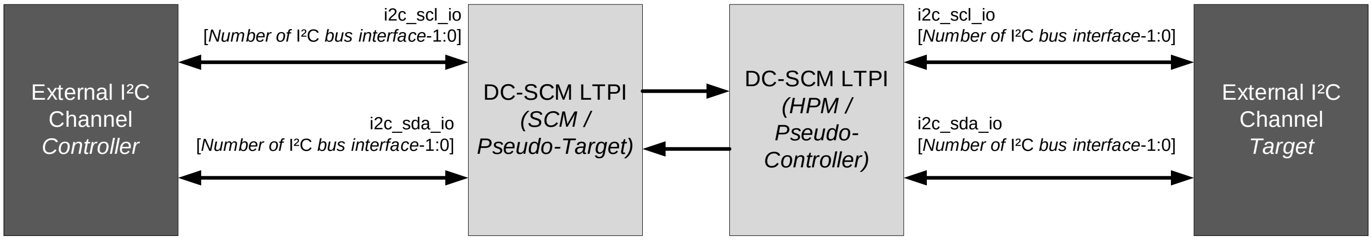


Figure 2.19. I^{2}C Pin List

Since I^{2}C bus is bidirectional in nature and handshaking is required between controller and target, I^{2}C channel mode must be set to either Controller or Target mode through *I^{2}C Channel Mode* attribute. When IP is meant to interface with an external I^{2}C controller, *I^{2}C Channel Mode* must be set to Controller. When IP is meant to interface with an external I^{2}C target, *I^{2}C Channel Mode* must be set to Target.

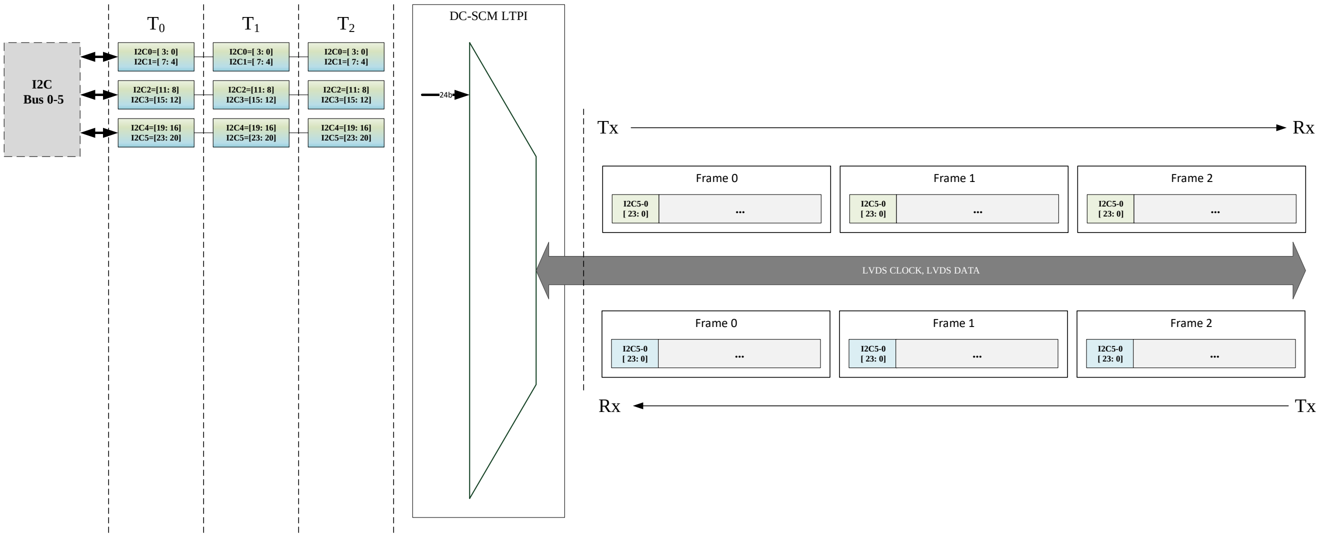


Figure 2.20. I^{2}C Mechanism

Same with other channels, payload allocation for both Rx and Tx frames depends on the *I²C Bus interface per Frame* 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 2.24. I²C Events Encoding

Bus Event	Starting Direction	Description	4-bit Event Encoding
Idle	Bi-directional	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	Bi-directional	SDA bit value recovered on remote side	0b0101
Data 0	Bi-directional	Send SDA bit value 0	0b0110
Data 1	Bi-directional	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	Bi-directional	Send SDA bit value 0 received correctly	0b1010
Data 1 Echo	Bi-directional	Send SDA bit value 1 received correctly	0b1011
Data Received Echo	Bi-directional	Indicates that Data Received event was correctly received. It is used to differentiate between consecutive Data 0/1 events during byte transmission.	0b1100
Reserved	Reserved	Reserved for future use. 0b1000	0b1100-0b1111

Each I²C bus is equivalent to 4-bit payload data as shown in Table 2.24. In the example in Table 2.23, for six I²C buses, three bytes of payload are allocated for I²C channel per frame correspondingly.

Table 2.25. 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]

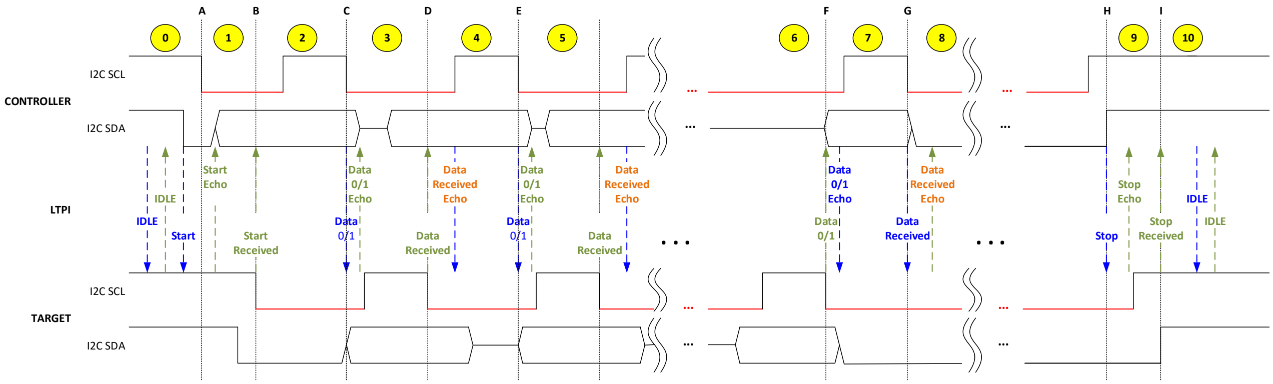


Figure 2.21. I²C Bus Event Exchange Between Controller and Target

Figure 2.21 shows the I²C bus event exchange. The echo support is required by the IP. 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 either 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. Red lines indicate the Clock Stretching done on SCL bus.

Table 2.26. 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
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

Phase	I^{2}C IP		LTPI	
	Controller	Target	Controller Event	Target Event
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
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				
…	…	…	…	…

Phase	I²C IP		LTPI	
	Controller	Target	Controller Event	Target Event
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
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

Phase	I²C IP		LTPI	
	Controller	Target	Controller Event	Target Event
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

2.9.2.1. 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, 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 (that is 400 MHz LVDS clock). IP with I²C channel enabled does not work if input system clock is less than 40 MHz. For list of I²C support limitation, refer to Appendix B.

2.9.3. UART Channel

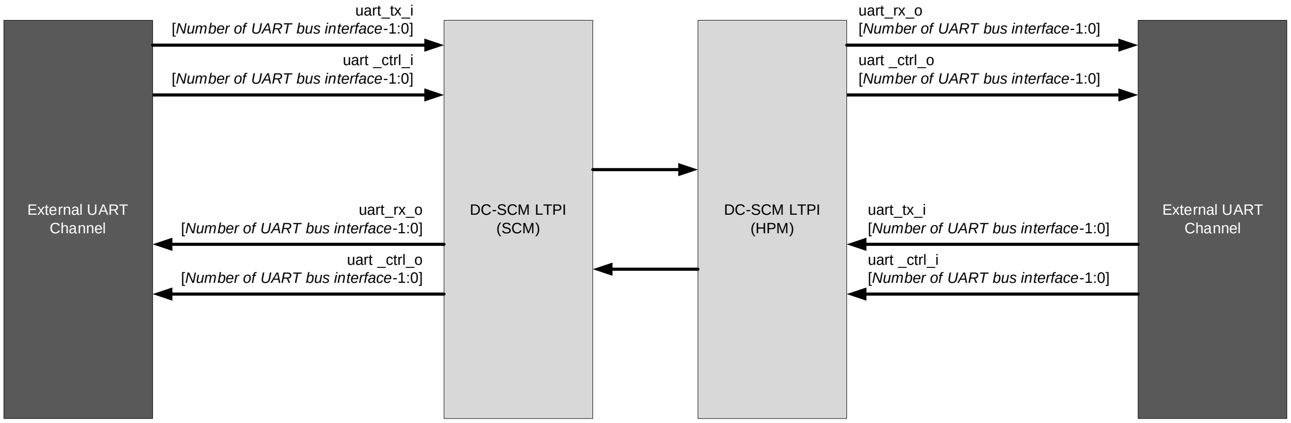

uart_tx_i
[Number of UART bus interface-1:0]
uart _ctrl_i
[Number of UART bus interface-1:0]
uart_rx_o
[Number of UART bus interface-1:0]
uart _ctrl_o
[Number of UART bus interface-1:0]
External UART Channel
DC-SCM LTPI (SCM)
DC-SCM LTPI (HPM)
uart_rx_o
[Number of UART bus interface-1:0]
uart _ctrl_o
[Number of UART bus interface-1:0]
uart_tx_i
[Number of UART bus interface-1:0]
uart _ctrl_i
[Number of UART bus interface-1:0]
External UART Channel

Figure 2.22. 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 3 and the three consecutive samples are being encoded in every LTPI Frame as shown in Figure 2.23.

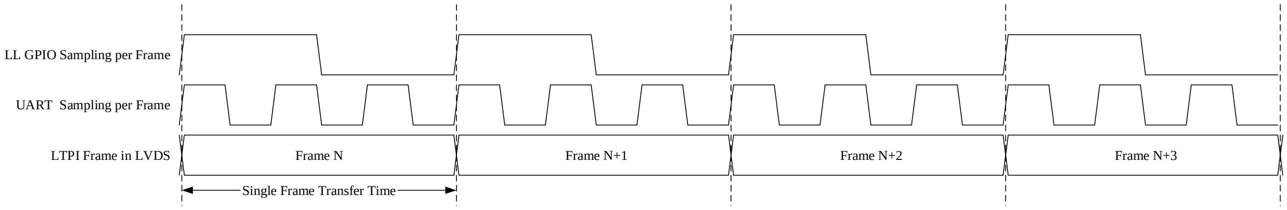


Figure 2.23. UART General Oversampling Principle

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 2.24. The same sampling distribution is implemented in regenerating the UART data.

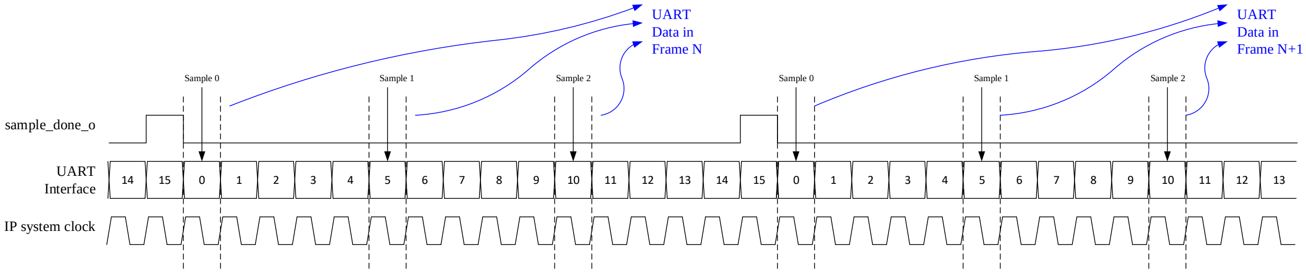


Figure 2.24. LTPI UART Sampling Distribution

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

Table 2.23 shows one byte is allocated for UART payload per frame, which is equivalent to two UART buses. Table 2.27 lists the content of each UART bus.

Table 2.27. 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 Tx, IP samples the UART input ports. If there is a total of 2 UART buses and *UART Bus interface per Frame* is set to 2, IP samples the two buses.

For the received frames, IP remaps the received payload into the UART output pins (i.e. 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.

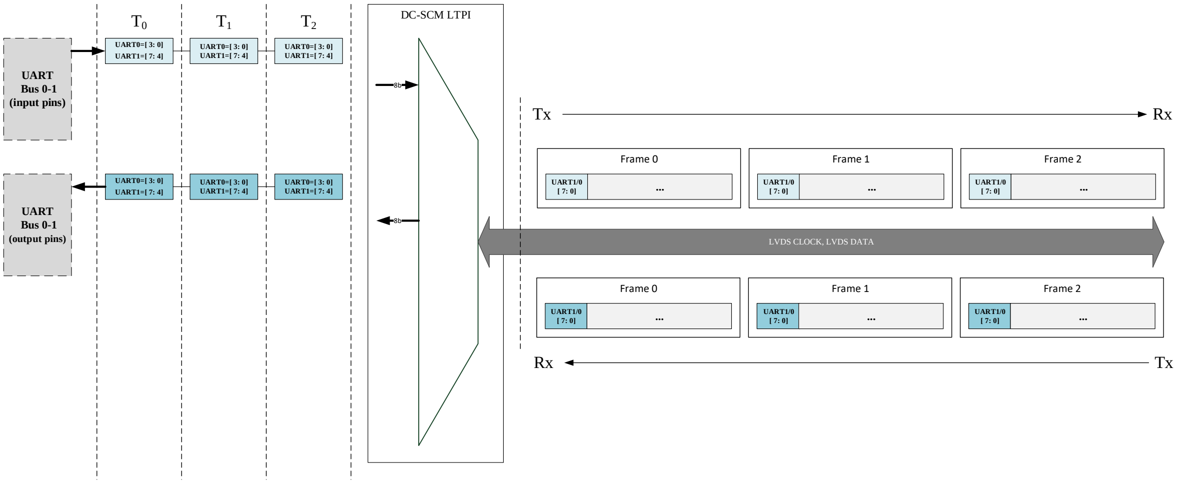


Figure 2.25. UART Mechanism

2.9.4. OEM Channel

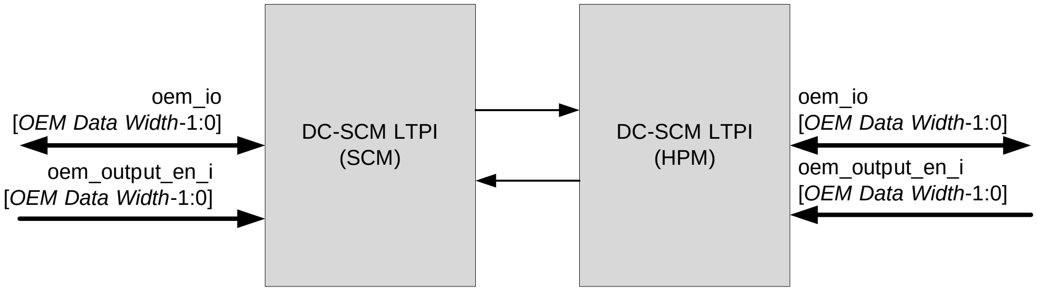


Figure 2.26. OEM Pin List

For OEM, pins are set as bidirectional and payload allocation for both Rx and Tx frames depends on the *OEM Payload Width per Frame* attribute set. It follows the sampling mechanism similar to LL GPIO except that when enabled, OEM interface is updated only every non-Data Channel frame transfer. IP samples the OEM ports based on *OEM Payload Width per Frame* attribute set. Table 2.23 shows the two bytes are allocated for OEM payload per frame. 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 the user to control the intended direction of the I/O through oem_output_en_i. When this port is high, I/O is set as an output. When this port is low, I/O is set as input.

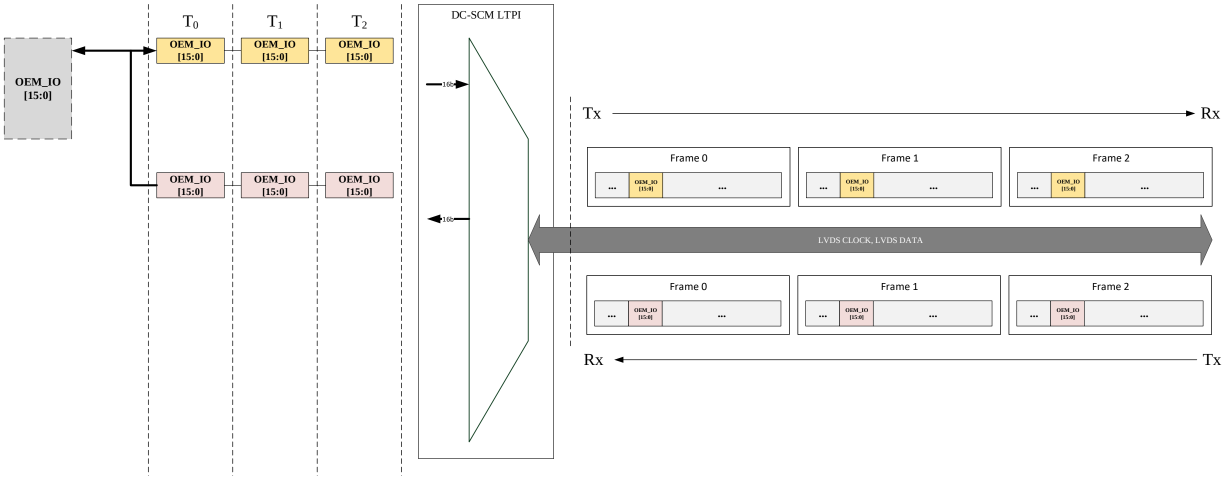


Figure 2.27. OEM Mechanism

2.9.5. Data Channel

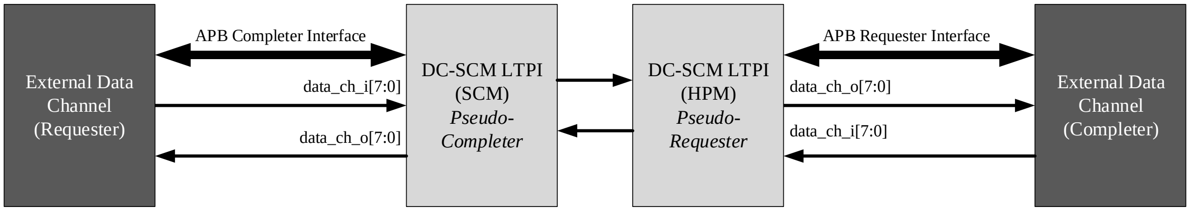

External Data Channel (Requester)
APB Completer Interface
data_ch_i[7:0]
data_ch_o[7:0]
DC-SCM LTPI (SCM) Pseudo-Completer
DC-SCM LTPI (HPM) Pseudo-Requester
APB Requester Interface
data_ch_o[7:0]
data_ch_i[7:0]
External Data Channel (Completer)

Figure 2.28. Data Channel Pin List

Data channel allows tunneling of standardized Read and Write requests to addressable memory spaces. DC-SCM LTPI soft IP supports APB interface for Data Channel 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.

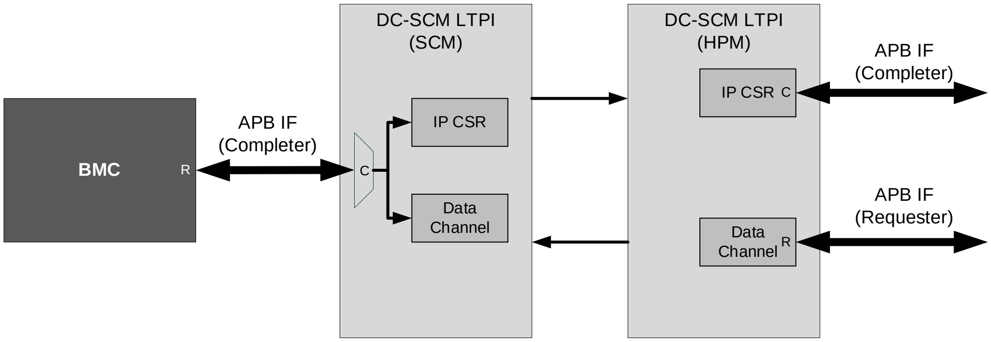


Figure 2.29. APB Interface for SCM and HPM

In SCM, Data channel can be initiated through APB access to register address starting from 0x00000400 onwards. In HPM, Data channel access is through a separate APB requester interface.

Table 2.28. Data Channel Command Encoding

Command	Parameters	Description	8-bit Event Encoding
Read Request	Address	Command indicates a request to read data from specific address.	0x00
Write Request	Address and Data	Command indicates a request to write data to a specific 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 operation won't 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 2.29. Memory Command Frame Mapping

Payload Byte Sequence	Map	Corresponding Value	Description
0	data_ch[7: 0]	data_ch_i[7:0]	Miscellaneous data channel signal. Tag field which can be used for customer specific purposes. Reconstructed to data_ch_o[7:0]. Only sampled when there is an active Data Channel transaction.
1	data_ch[15: 8]	Command	8-bit Command Encoding. Refer to Table 2.28.
2	data_ch[23:16]	Address Byte 3	Address for the request as indicated in the APB interface. For CRC Error command frame, this takes last successful known address.
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 2.30.
7	data_ch[63:56]	Data Byte 3	Data bytes for Write Request and Read Completion commands as indicated in the APB interface. 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 2.30. 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 tag field described in Table 2.29 is generated from data_ch_i and is reconstructed to data_ch_o of the remote IP. It is sampled and updated once every active Data channel access in APB interface. Definition and/or information that this field carries is outside of scope of the IP and is customer specific. In cases when this field is not intended to be used, this must be tied to a known value.

The time it takes to finish a complete end-to-end APB transaction varies depending on the APB clock frequency used and target device response. 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. In a simple case with external APB completer device connected to HPM implementing a NO WAIT response mechanism, it might take approximately 10-15 LTPI frames to finish a complete transaction.

The sample use of Data Channel is shown in Figure 2.30 and the timing diagram of a complete end-to-end transaction is shown in Figure 2.31.

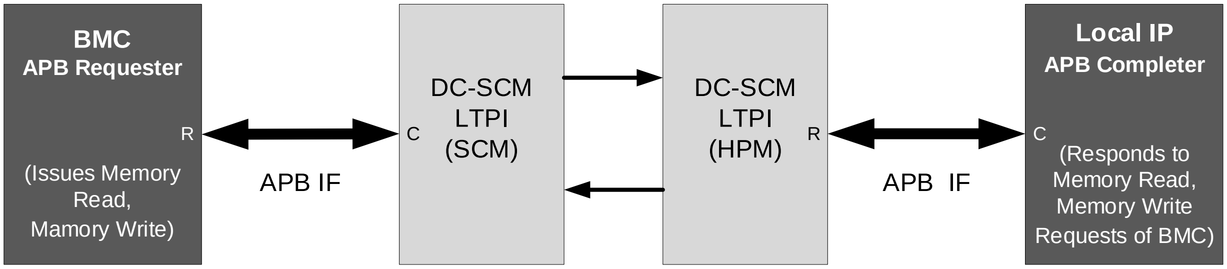

BMC
APB Requester
R
(Issues Memory Read, Mamory Write)
APB IF
C
DC-SCM LTPI (SCM)
DC-SCM LTPI (HPM)
R
APB IF
C
Local IP
APB Completer
(Responds to Memory Read, Memory Write Requests of BMC)

Figure 2.30. Example of Data Channel Usage

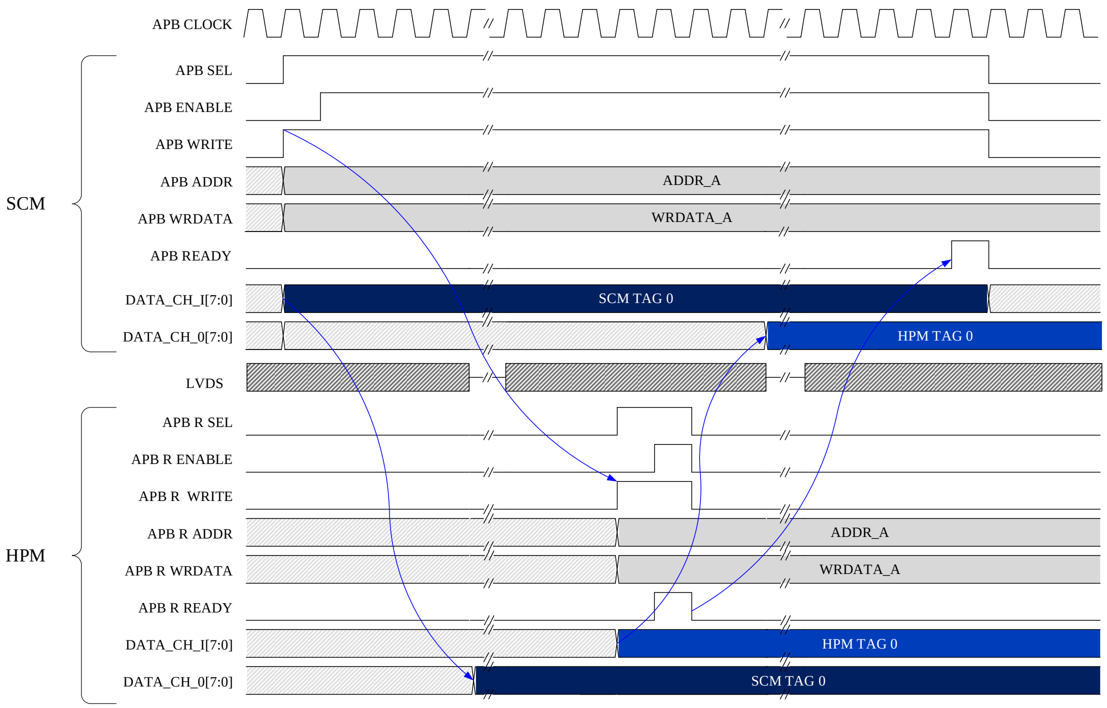


Figure 2.31. Sample SCM to HPM APB Transaction

APB completer interface in SCM receives Data Channel transaction request (that is write operation) from the connected external APB requester. SCM processes the transaction and holds APB_READY to LOW to extend the transfer. SCM encodes the write request to LTPI Data Channel command and transmits the frame to HPM through LTPI. HPM decodes the received data channel frame and drives the APB requester interface accordingly. APB requester interface in HPM waits for the response of the external APB completer device. Once external APB completer device drives APB_R_READY to HIGH, HPM generates a completion frame and sends it back to SCM through LTPI. SCM decodes the received data channel frame and drives the APB completer interface accordingly by toggling APB_READY.

The internal registers of the IP set as HPM can also be accessed through Data Channel, but it is up to the user to connect the APB completer interface of the HPM IP to the appropriate bus interconnect. Figure 2.32 shows the sample block diagram.

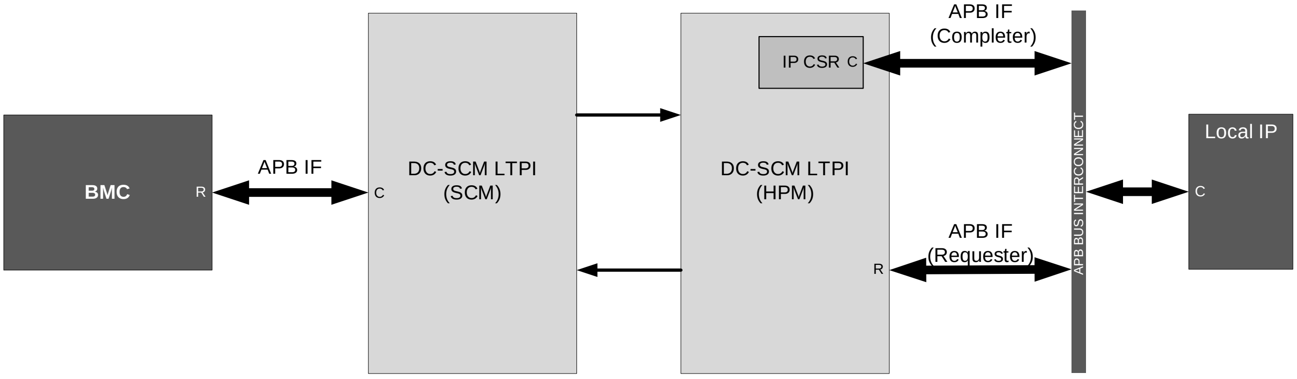


Figure 2.32. Sample HPM IP CSR Access Through Data Channel Block Diagram

2.9.5.1. 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 make a decision of retransmission or return error to the originator of Data Channel access. Table 2.31 lists the summary of how IP processes CRC error for Data Channel.

Table 2.31. 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 dc_err_o to indicate that previous completion data frame transaction from HPM is not received correctly by SCM.
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 (i.e. Memory Write/Read request command). • HPM drops the frame and generates Data Channel command indicating CRC error (i.e. 0x04 command).

2.10. Clocks and Reset

Below are the list of clocks used by the IP. Figure 2.33 shows the general clock topology.

- Sync clock (sync_clk_i) is used for internal GDDR clock synchronization. This clock should be the equal or less than the slowest clock in the system and is asynchronous with respect to the other clocks of the IP.
- Edge clock (eclk_i) and 90 degree phase shifted edge clock (eclk90_i) are clocks used by the GDDR Tx 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.
- 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.
- Both DDR Tx and Rx blocks generate core clock (sclk) 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.
- A system clock (clk_i) is used to clock the write operation of Tx FIFO and read operation of the Rx FIFO. This clock is also used to clock the transmit and receive related logic.

It is assumed that external channels are operating at very much lower frequencies compared to the operating system clock of the IP.

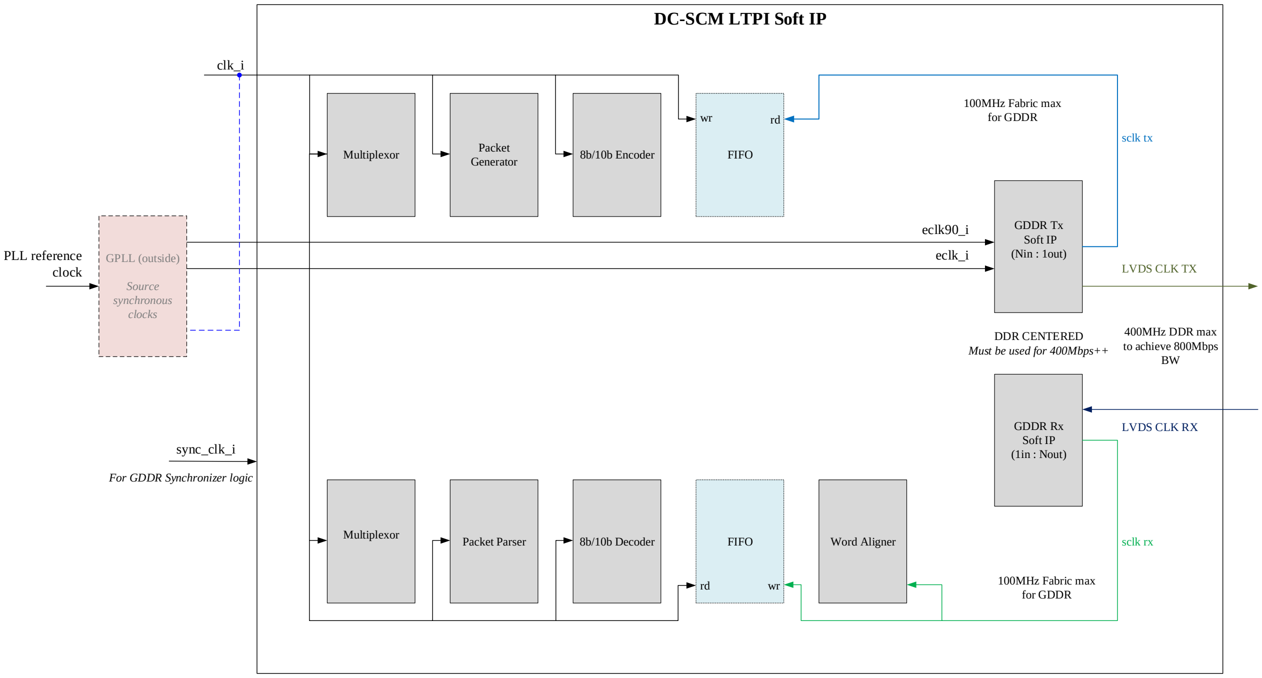


Figure 2.33. General Clocking Topology (Main)

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.

To avoid FIFO underflow/overflow in the Tx and Rx domains, system clock must match the input/output throughput of the FIFO. In system clock's perspective, data width of FIFO is always 10-bit wide. Suggested clocking implementation is shown in Figure 2.34.

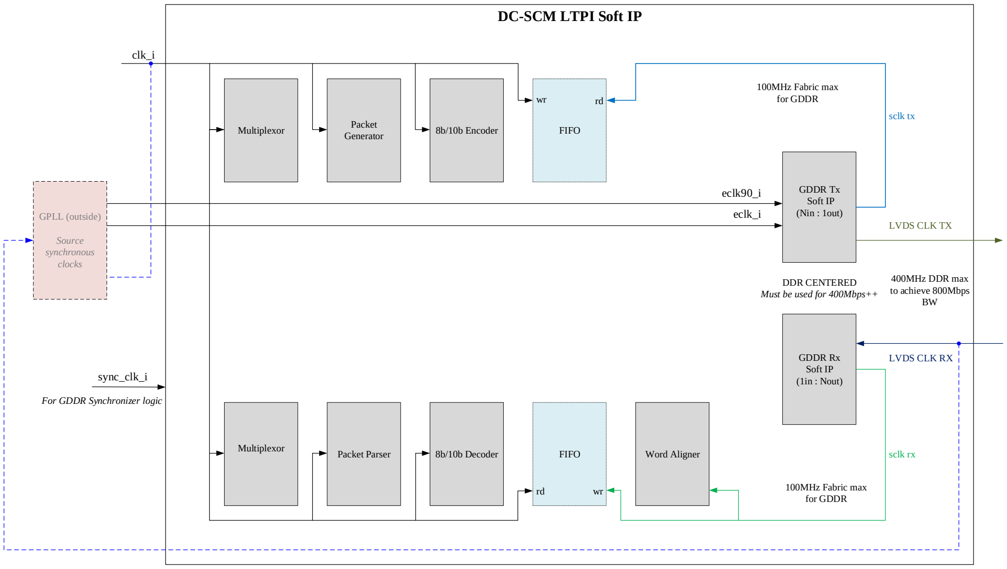

DC-SCM LTPI Soft IP
clk_i
Multiplexor
Packet Generator
8b/10b Encoder
wr
rd
FIFO
100MHz Fabric max for GDDR
sclk tx
GPLL (outside)
Source synchronous clocks
eclk90_i
eclk_i
GDDR Tx Soft IP (Nin : 1out)
LVDS CLK TX
DDR CENTERED
Must be used for 400Mbps++
400MHz DDR max to achieve 800Mbps BW
GDDR Rx Soft IP (1in : Nout)
LVDS CLK RX
sync_clk_i
For GDDR Synchronizer logic
Multiplexor
Packet Parser
8b/10b Decoder
FIFO
Word Aligner
sclk rx
rd
wr
100MHz Fabric max for GDDR

Figure 2.34. Suggested Clocking Implementation (Follower)

Either SCM or HPM should be set as the main source of clock and has an independent clocking system. The other module uses the incoming LVDS Rx clock as reference clock of the clock generation logic (that is PLL) and serves as clock follower. Implementing such ensures that clocks are properly synchronized between SCM and HPM.

Figure 2.35 shows the Device 1 (SCM) is set as the main source of clock and has independent PLL reference clock sourcing. Device 2 (HPM) uses the LVDS clock coming from Device 1 as reference clock of its PLL. Device 2 serves as the clock follower. 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.

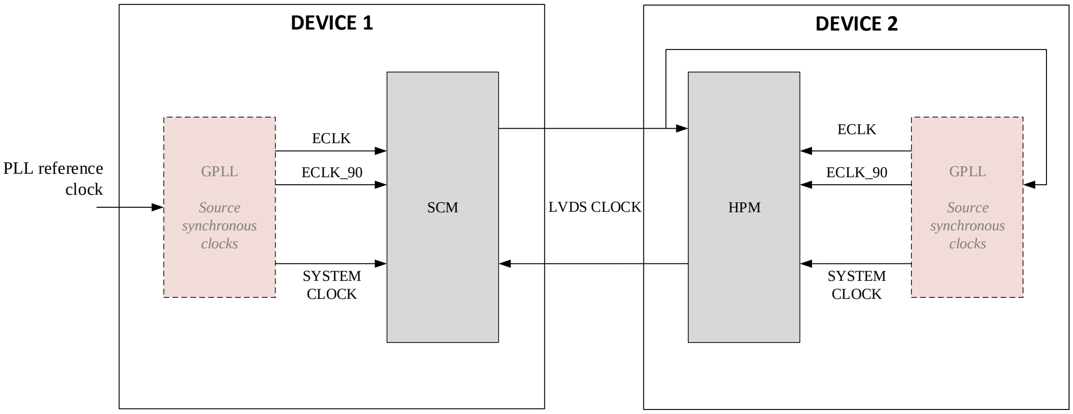


Figure 2.35. End to End Clocking Implementation

2.10.1. Clock Computations

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

$$DDR\ Gear = 8\ (MachXO3)$$

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

$$Edge\ Clock\ = LVDS\ Clock$$

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

$$CLK\_I = PHY\ Data\ Rate\ /\ 10$$

2.10.2. 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
- 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

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

- 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

2.10.3. Clock Compensation

For special cases that recommended clocking implementation is not followed, IP supports an optional frame drop and frame repeat to compensate for clock difference during FIFO underflow and overflow. This is enabled through *Enable Clock Compensation* attribute. For FIFO overflow, IP may drop up to two frames until FIFO recovers. For IP underflow, IP may send up to two dummy frames to fabric logic until FIFO recovers.

In cases that FIFO performs frame drop or frame repeat, IP may encounter expected CRC error and internal latency from frame logic to PHY and vice versa might increase. For cases that link-lost is encountered due to significant clock difference in the FIFO read and write operations, it is outside of scope of the IP to perform recovery.

2.10.4. Reset

Asynchronous active low reset (reset_n_i) with synchronous release is implemented as system reset of the IP. It is recommended to implement a reset synchronizer circuit in high-level/system design when this IP is used, and reset_n_i must be synchronized to the input system clock (clk_i).

2.11. Error Impact, Handling, and Recovery

Table 2.32. IP Error Handling

Scenario	Tx	Rx
Failed Word Alignment	Tx assumes that receiving host can finish link synchronization/word alignment within the training period set.	Error handling is out of scope by the IP. IP issues flag/interrupt to indicate if word alignment is achieved.
LTPI FSM protocol timeout	IP issues corresponding timeout flag signal to indicate timeout is encountered.	Error handling is out of scope by the IP. IP issues flag/interrupt to indicate timeout.
CRC Error	N/A	IP issues flag/ interrupt to indicate failure. The received frame is discarded once Rx detects wrong CRC in the received frame. For each channel, CRC error impact is different. Refer to Table 2.33 for more details.

Table 2.33. 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.

Channels	Impact	Recovery
I^{2}C	The state of I^{2}C is maintained based on previous successful frame. The next frame with correct CRC provides the updated states of the I^{2}C. In I^{2}C perspective, this condition does not yet affect the local bus since clock stretching is implemented. As long as CRC error is not permanent, the correct I^{2}C 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 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 make a decision of retransmission or return error to the originator of Data Channel access. Refer to Table 2.31 for details.	Not needed if CRC error is not permanent. If error is permanent, the link is lost and re-initialization is required.

Table 2.34. Timeout Impact on LTPI Channels

Channels	Impact	Recovery
Low Latency GPIO	N/A	N/A
Normal Latency GPIO	N/A	N/A
I^{2}C	If LTPI events are not received by the other end, I^{2}C stays in its previous state.	None. I^{2}C in nature does not have a timeout. It is up to the user to implement a timeout monitor/logic and terminate the current I^{2}C transaction in case bus events are not observed within user's given timeout/perspective. IP provides a separate I^{2}C channel reset through I2C_BUS_RST register to reset the internal I^{2}C controller of the IP to recover the bus in case of timeout/errors.
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 the user to terminate the current APB Data channel transaction in case events are not observed within user's given timeout/perspective. IP provides a separate APB channel reset through IP_CTRL register to reset the internal Data Channel controller of the IP to recover the bus in case of timeout/errors.

Appendix A. Resource Utilization

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

Table A.1. Resource Utilization

Configuration	Registers	Slice	LUTs	EBRs
LCMXO3L-9400C-6BG484C IP Mode = SCM DDR == Enabled Low Latency I/O == 16 Normal Latency I/O == 16 I²C == 1 (Controller) UART == 2 OEM == 16 Data Channel == Enabled	2346	1202	2296	6
LCMXO3L-9400C-6BG484C IP Mode = HPM DDR == Enabled Low Latency I/O == 16 Normal Latency I/O == 16 I²C == 1 (Target) UART == 2 OEM == 32 Data Channel == Enabled	2237	1157	2235	6

Note: Utilization data is generated using a sample IP configuration for LCMXO3L-9400C-6BG484C with Strategy set to default setting. Number may vary when using a different software version or targeting a different device density, synthesis tool, or speed grade. For Static Timing Analysis, IP is tested in both -5 and -6 speed grade. For better performance in certain cases, user is recommended to run multiple iterations of Place and Route and/or set Optimization Goal to *Timing* in Strategy Section of SW tool.

Appendix B. Limitations

The following lists the I^{2}C channel supports:

- I^{2}C tunneling support is limited to Standard (100 kHz) and Fast (400 kHz) modes only
- Echo support is always enabled

References

For complete information on Lattice Project-Based Environment, Design Flow, Implementation Flow and Tasks, as well as on the Simulation Flow, see the Lattice Propel Builder User Guide.

- MachXO3 FPGA-Web Page in latticesemi.com
- Lattice Memory Mapped Interface and Lattice Interrupt Interface User Guide (FPGA-UG-02039)

Technical Support Assistance

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

Revision History

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^{2}C group. - Removed *Data Frame Type == Custom* from the condition in some of the attributes and added two rows for new attributes in the UART group. - Updated Table 2.3. Attributes Description to add two rows for new attributes. - Updated Table 2.4. DC-SCM LTPI Soft IP Registers to change default value of 0x64 to 24'h0.

Revision 1.0, June 2022

Section	Change Summary
All	Initial release.