



Gowin DC-SCM LTPI IP

User Guide

IPUG1086-1.0.1E, 06/13/2025

Copyright © 2025 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is the trademark of Guangdong Gowin Semiconductor Corporation and is registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
02/08/2024	1.0E	Initial version published.
06/13/2025	1.0.1E	“Figure 4-1 Typical Clock Scheme” updated.

Contents

Contents	i
List of Figures	iii
List of Tables	iv
1 About This Guide	1
1.1 Purpose	1
1.2 Related Documents	1
1.3 Terminology and Abbreviations	1
1.4 Support and Feedback	2
2 Function Overview	3
2.1 Overview	3
2.2 Features	4
2.3 Resource Utilization	4
3 Functional Description	6
3.1 System Block Diagram	6
3.1.1 DC-SCM 2.0 LTPI Architecture	6
3.1.2 IP Implementation Block Diagram	7
3.2 Port List	9
3.3 Interface Timing	12
3.3.1 Configuration Interface Timing	12
3.3.2 Data Interface Timing	12
3.3.3 GPIO Interface Timing	12
3.3.4 UART Interface Timing	12
3.3.5 IIC/SMBus Interface Timing	12
3.4 Register Description	13
4 Design Description	18
4.1 Line Rate	18
4.2 Clock	18
4.2.1 System Clock	18
4.2.2 Reference Clock	18
4.2.3 Configuration Clock	18
4.2.4 LVDS Clock	18

4.2.5 RX/TX DIV5 Clock	19
4.2.6 Clock Structure	19
4.3 External Channel Interface Processing	19
4.3.1 GPIO Channel	20
4.3.2 UART Channel.....	21
4.3.3 IIC/SMBus Channel	23
4.3.4 OEM Channel	27
4.3.5 Data Channel.....	27
4.4 LTPI Frame Format.....	29
4.4.1 Frame Format.....	29
4.4.2 Specific Frame Definition.....	29
4.5 Link Initialization & Operation Flow	34
4.5.1 State Diagram	34
4.5.2 Link Training	35
4.5.3 Link Configuration.....	36
4.5.4 Link Operating	37
5 Interface Configuration	38
6 Reference Design	43
6.1 Application	43
6.2 Reference Design	43
6.2.1 Hardware Platform.....	43
6.2.2 Implementation Diagram	44
6.2.3 Bus Protocol & Address	45
6.3 On-Board Testing	46
6.3.1 Operation Description	46
6.3.2 Operating Flow	46
7 File Delivery	50
7.1 Documentation.....	50
7.2 Design Source Code (Encryption)	50
7.3 Reference Design	50

List of Figures

Figure 3-1 LTPI Architecture Diagram.....	6
Figure 3-2 LTPI Tunneling Diagram	7
Figure 3-3 DC-SCM LTPI IP Implementation Block Diagram	7
Figure 3-4 Gowin DC-SCM LTPI IP Port Diagram	9
Figure 3-5 Write Timing Diagram of Configuration Interface	12
Figure 3-6 Read Timing Diagram of Configuration Interface	12
Figure 4-1 Typical Clock Scheme	19
Figure 4-2 UART Sampling and Recovery Diagram	22
Figure 4-3 IIC/SMBus Bus Transaction Switch.....	24
Figure 4-4 LTPI Link State Diagram.....	34
Figure 5-1 Open DC-SCM LTPI IP.....	38
Figure 5-2 Gowin DC-SCM LTPI IP Configuration Interface 1.....	39
Figure 5-3 Gowin DC-SCM LTPI IP Configuration Interface 2.....	40
Figure 5-4 Gowin DC-SCM LTPI IP Configuration Interface 3.....	40
Figure 5-5 Gowin DC-SCM LTPI IP Configuration Interface 4.....	41
Figure 5-6 Gowin DC-SCM LTPI IP Configuration Interface 5.....	41
Figure 5-7 Gowin DC-SCM LTPI IP Configuration Interface 6.....	41
Figure 5-8 Gowin DC-SCM LTPI IP Configuration Interface 7.....	42
Figure 5-9 Gowin DC-SCM LTPI IP Configuration Interface 8.....	42
Figure 6-1 HPM and SCM Connection	43
Figure 6-2 Platform Block Diagram.....	43
Figure 6-3 Implementation Diagram of Reference Design	44

List of Tables

Table 1-1 Terminology and Abbreviations	1
Table 2-1 Gowin DC-SCM LTPI IP Overview	3
Table 2-2 Gowin DC-SCM LTPI IP Resource Utilization	4
Table 3-1 Port Definition	10
Table 3-2 Gowin DC-SCM LTPI IP Register	13
Table 3-3 Speed Description	16
Table 3-4 Advertisement Frame Capability Description	16
Table 3-5 Maximum Baud Rate of Serial Port Channels	17
Table 4-1 External Channel Interface Processing	20
Table 4-2 GPIO Field Configuration	21
Table 4-3 UART Field Configuration	22
Table 4-4 IIC/SMBus Channel Transaction Definition	23
Table 4-5 IIC/SMBus Field Configuration	24
Table 4-6 IIC/SMBus Transaction Flow	25
Table 4-7 Data Channel Transaction Definition	27
Table 4-8 Data Channel Field Configuration	28
Table 4-9 Frame Format Description	29
Table 4-10 Link Training Phase Frame Type	29
Table 4-11 Link Detect Frame Definition	29
Table 4-12 Link Speed Frame Definition	30
Table 4-13 Link Configuration Phase Frame Type	30
Table 4-14 Advertise Frame Definition	30
Table 4-15 Configure Frame Definition	31
Table 4-16 Accept Frame Definition	32
Table 4-17 Operation Phase Frame Type	32
Table 4-18 Default I/O Frame Definition	33
Table 4-19 Default Data Frame Definition	33
Table 6-1 Key and LED Mapping Route	45
Table 6-2 Register Definition	45
Table 7-1 Documents List	50
Table 7-2 Gowin DC-SCM LTPI IP Design Source Code List	50
Table 7-3 Gowin DC-SCM LTPI IP RefDesign Folder Content List	50

1 About This Guide

1.1 Purpose

The purpose of Gowin DC-SCM LTPI IP User Guide is to help you learn the features and usage of Gowin DC-SCM LTPI IP by providing the descriptions of functions, GUI, and reference design, etc. The software screenshots and the supported products listed in this manual are based on Gowin Software 1.9.9.01 (64-bit). As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

1.2 Related Documents

The latest user guides are available on the GOWINSEMI Website. You can find related documents at www.gowinsemi.com:

- [SUG100, Gowin Software User Guide](#)
- [DS961, GW2ANR series of FPGA Products Data Sheet](#)
- [DS102, GW2A series of FPGA Products Data Sheet](#)
- [DS226, GW2AR series of FPGA Products Data Sheet](#)
- [DS971, GW2AN-18X & 9X Data Sheet](#)
- [DS976, GW2AN-55 Data Sheet](#)

1.3 Terminology and Abbreviations

Table 1-1 shows the abbreviations and terminology used in this manual.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
APB	Advanced Peripheral Bus
CRC	Cycle Redundancy Check
CSR	Control and Status Register
DC-SCI	Datacenter-ready Secure Control Interface
DC-SCM	Datacenter-ready Secure Control Module
FPGA	Field Programmable Gate Array

Terminology and Abbreviations	Meaning
GPIO	Gowin Programmable Input/Output
HPM	Host Processor Module
IIC	Inter-Integrated Circuit
IP	Intellectual Property
LPTI	LVDS Tunneling Protocol and Interface
LVDS	Low Voltage Differential Signaling
MCSI	Multi-Channel Serial Interface
OEM	Original Equipment Manufacturer
SCM	Secure Control Module
SerDes	Serializer/Deserializer
SMBus	System Management Bus
TDM	Time Division Multiplexing
UART	Universal Asynchronous Receiver/Transmitter

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Function Overview

2.1 Overview

LVDS Tunneling Protocol & Interface (LTPI) is a protocol and interface designed to transfer various low-speed signals between Host Processor Module (HPM) and Secure Control Module (SCM). LTPI supports tunneling of not only GPIO but also low-speed serial interfaces such as IIC and UART. LTPI can also be extended with additional proprietary OEM interfaces and provide support for raw data tunneling between HPM FPGA and SCM FPGA. Gowin DC-SCM LTPI IP complies with the Datacenter-Ready Secure Control Module (DC-SCM) 2.0 LTPI specification and features a standardized Datacenter-Ready Secure Control Interface (DC-SCI). The IP aggregates multiple data channels, such as IIC/SMBbus, GPIO, and UART, increasing more flexibility to customers' systems and circuit board designs.

Table 2-1 Gowin DC-SCM LTPI IP Overview

Gowin DC-SCM LTPI IP	
Logic Resource	See Table 2-2.
Delivered Doc.	
Design Files	Verilog (encrypted)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	GowinSynthesis
Application Software	Gowin Software (V1.9.9.Beta-1 and above)

Note!

For the devices supported, you can click [here](#) to get the information.

2.2 Features

- Compliant with DC-SCM 2.0 Specification
- Support link initialization, discovery, and negotiation
- Support multi-channel serial interface
- Support GPIO, IIC/SMBus, UART, OEM, and data channel aggregation
- Support aggregation/disaggregation of up to 5 channels
- Support normal latency GPIO up to 1023
- Each channel of the IIC/SMBus can be configured as Controller or Target
- OEM channels are extended to low-latency GPIO interfaces
- Support AMBA 3 APB Protocol Specification V1.0 for register access of soft IP and data channels
 - Support 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)
- Support LVDS data rate for GW2A devices up to 300Mbps (Preliminary)

2.3 Resource Utilization

Gowin DC-SCM LTPI IP can be implemented by Verilog. Its performance and resource utilization may vary when the design is employed in different devices, or at different density, speed, or grade. Taking Gowin GW2A series of FPGA as an instance, the resource utilization of Gowin DC-SCM LTPI IP is as shown in Table 2-2.

Table 2-2 Gowin DC-SCM LTPI IP Resource Utilization

Configuration	Resource Utilization					
	Register	LUT	ALU	BSRAM	SSRAM	PLL
Configure as SCM Enable GPIO, IIC, UART, OEM, and data channel LLGPIO is set to 16 NLGPIO is set to 32 UART interface is set to 1 IIC interface is set to 1 OEM GPIO is set to 32	2535	2708	581	0	0	1
Configure as HPM Enable GPIO, IIC, UART, OEM, and data channel LLGPIO is set to 16 NLGPIO is set to 32	2611	2794	596	0	1	1

Configuration	Resource Utilization					
	Register	LUT	ALU	BSRAM	SSRAM	PLL
UART interface is set to 1 IIC interface is set to 1 OEM GPIO is set to 32						

3 Functional Description

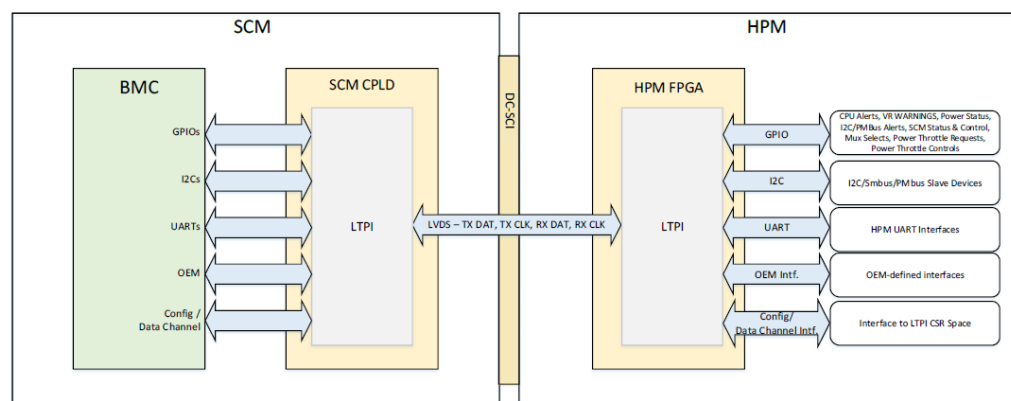
3.1 System Block Diagram

3.1.1 DC-SCM 2.0 LTPI Architecture

The LTPI interface can be implemented with two FPGA devices for configuration with typical BMC devices:

- HPM FPGA: Provide a bridging between local HPM interface and LTPI
- SCM FPGA: Provide a bridging between local SCM interface and LTPI

Figure 3-1 LTPI Architecture Diagram



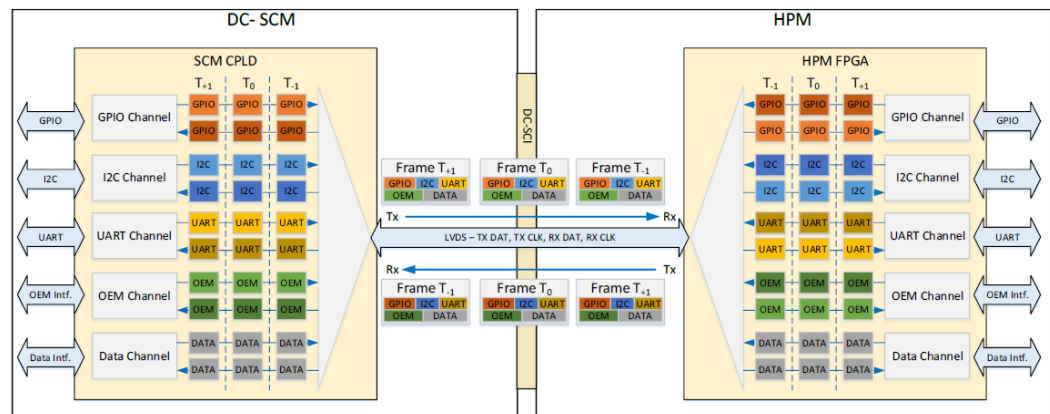
The DC-SCM LTPI interface implements the concept of channels, where each channel is mapped to a specific type of physical interface on the DC-SCM and HPM. The following channels are defined for the LTPI interface:

- GPIO channel
- IIC/SMBus channel
- UART channel
- OEM channel
- Data channel

LTPI uses Time Division Multiplexing (TDM) of a high-speed LVDS full-duplex link to transmit and receive LTPI channels between SCM and

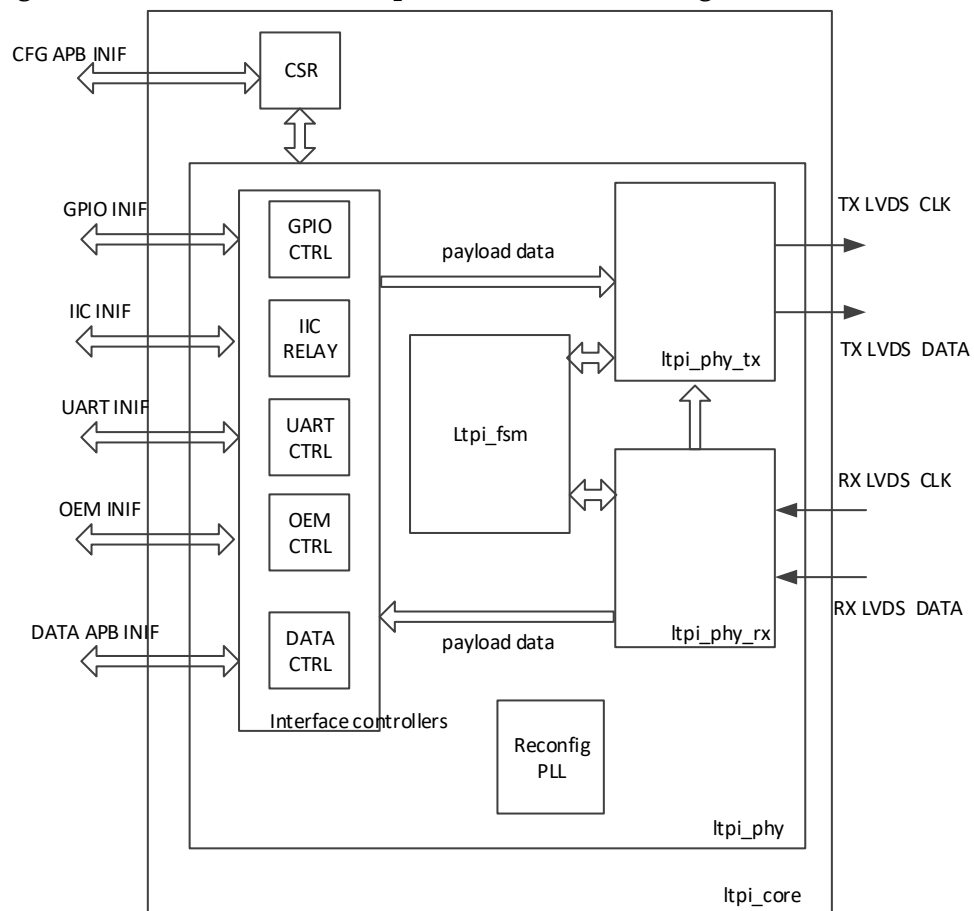
HPM. There is one LTPI frame being transmitted in each equal time slot on the LVDS link, as shown in Figure 3-2. In each frame sent over an LTPI interface, the number of bits allocated to a specific channel is proportional to the LTPI bandwidth dedicated to each channel. To minimize the latency effect on the LTPI channel, it is assumed that LTPI frames have no inter-frame gap. In implementations where inter-frame gaps cannot be avoided, it is recommended that the inter-frame gap time remain constant to minimize latency variations and simplify the implementation.

Figure 3-2 LTPI Tunneling Diagram



3.1.2 IP Implementation Block Diagram

Figure 3-3 DC-SCM LTPI IP Implementation Block Diagram



The DC-SCM LTPI IP includes:

- Interface Controller Module: Receive signals from the external interface, capture or perform transactional encoding through the various internal controllers/repeaters, send the captured and encoded data to the frames in `ltpi_phy_tx` module; receive the contents of the operational frames from the LTPI, which are restored to the signal timing of the external interface. The module contains the following interfaces:
 - IIC/SMBus repeater-IIC/SMBus interface
 - GPIO Controller-UART interface
 - UART controller-UART interface
 - OEM controller-OEM interface
 - Data bus controller-data channel bridged to external interfaces
- `ltpi_phy_tx` Module Function
 - Each LTPI channel is encoded on dedicated bit fields within the LTPI frame.
 - Calculate the CRC of each frame and place it at the end of the frame.
 - Then encode the frame with 8b/10b encoding.
 - Frame flag is extended with frame comma symbol (K-character).
 - 10b encoding frames are serialized on the LVDS TX link.
- `ltpi_phy_rx` Module Function
 - LVDS traffic is captured and deserialized into a 10b encoding FIFO buffer.
 - In the 10b data stream, the frame K-character is captured as the start of the LTPI frame.
 - The frame is decoded as an 8b encoding.
 - Check the CRC checksum of the frame again and break it down into LTPI channels.
 - Rebuild channel state on each channel with the channel controller/repeater.
- `ltpi_fsm` Module Function
 - Initialize links and run process state machines.
- Reconfig PLL Module Function
 - Based on the rate negotiation results of SCM and HPM, adjust PLL parameters to generate the corresponding LVDS rate.

3.2 Port List

The IO ports of Gowin DC-SCM LTPI IP are shown in Figure 3-4 and the detailed descriptions are shown in Table 3-1.

Figure 3-4 Gowin DC-SCM LTPI IP Port Diagram

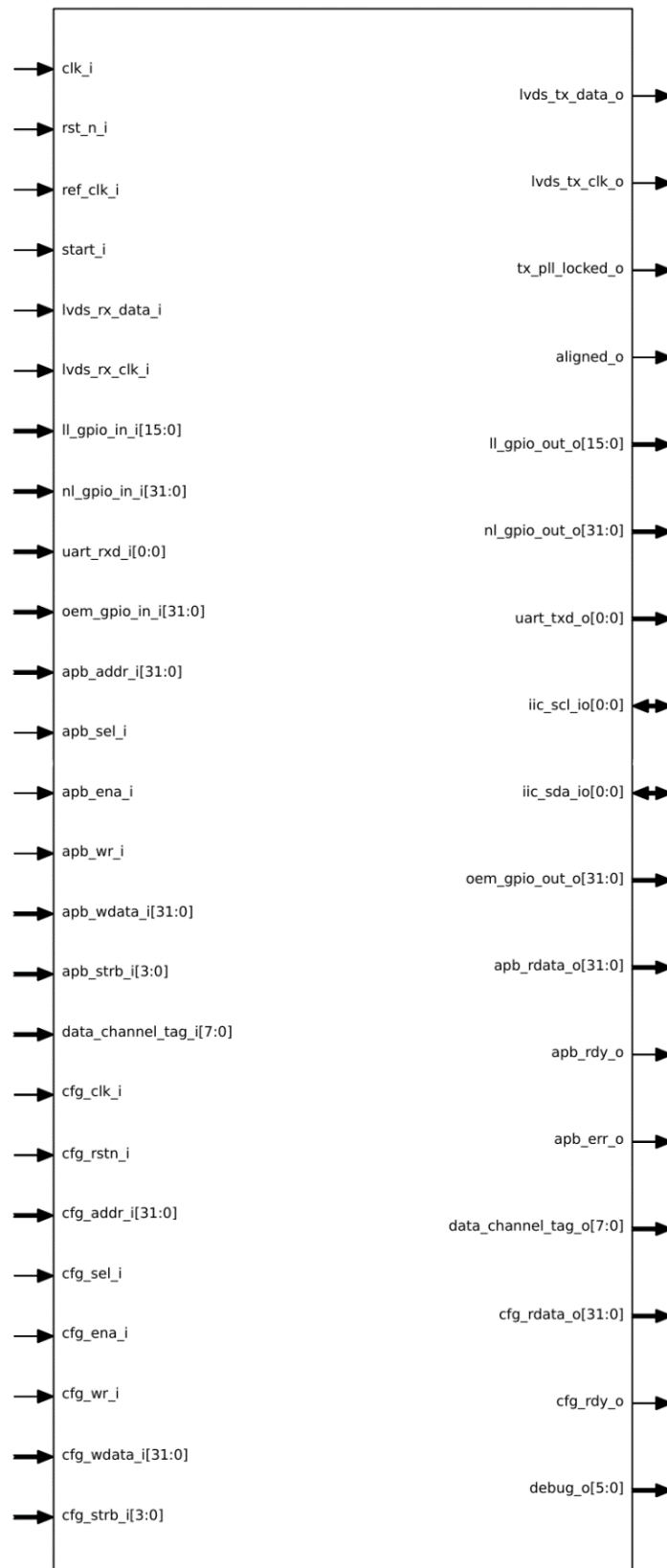


Table 3-1 Port Definition

Name	I/O	Data Width	Description
clk_i	input	1	System clock which must be 60MHz
rst_n_i	input	1	System Reset
ref_clk_i	input	1	Reference clock which must be 25MHz
start_i	input	1	IP core start working signal When it is high, the IP starts running.
lvds_tx_data_o	output	1	LTPI serial output data
lvds_tx_clk_o	output	1	LTPI serial output clock
lvds_rx_data_i	input	1	LTPI serial input data
lvds_rx_clk_i	input	1	LTPI serial input clock
tx_pll_locked_o	output	1	Internal PLL lock flag, which drives the TX differential clock and data
aligned_o	output	1	Byte alignment flag
Low-latency GPIO channel (Valid when GPIO interface is enabled)			
ll_gpio_in_i	input	[N ^[1] -1 : 0]	Low-latency GPIO input pin
ll_gpio_out_o	output	[N-1 : 0]	Low-latency GPIO output pin
Normal-latency GPIO channel (Valid when GPIO interface is enabled)			
nl_gpio_in_i	input	[M ^[2] -1 : 0]	Normal-latency GPIO input pin
nl_gpio_out_o	output	[M-1 : 0]	Normal-latency GPIO output pin
UART channel (Valid when UART interface is enabled)			
uart_rxd_i	input	[L ^[3] -1 : 0]	UART input pin
uart_cts_i	input	[L-1 : 0]	UART flow control pin
uart_txd_o	output	[L-1 : 0]	UART output pin
uart_rts_o	output	[L-1 : 0]	UART flow control pin
IIC/SMBus channel (Valid when IIC/SMBus interface is enabled)			
iic_scl_io	inout	[K ^[4] -1 : 0]	IIC/SMBus SCL pin
iic_ssda_io	inout	[K-1 : 0]	IIC/SMBus SDA pin
OEM channel (Valid when OEM interface is enabled)			
oem_gpio_in_i	input	[P ^[5] -1 : 0]	OEM input pin
oem_gpio_out_o	output	[P-1 : 0]	OEM output pin
Data channel (Valid when data interface is enabled)			
data_channel_tag_i	input	[7:0]	Tag field content input for data channel frame
data_channel_tag_o	output	[7:0]	Tag field content output for data channel frame
Data channel APB slave (SCM), (Valid when data Interface is enabled)			
apb_addr_i	input	[31:0]	Bus configuration address
apb_sel_i	input	1	Bus configuration selection
apb_ena_i	input	1	Bus configuration enable
apb_wr_i	input	1	Bus I/O signal 1: Write 0: Read
apb_rdata_o	output	[31:0]	Read data
apb_wdata_i	input	[31:0]	Write data

Name	I/O	Data Width	Description
apb_rdy_o	output	1	Ready signal
apb_err_o	output	1	Error indication
apb_strb_i	input	[3 :0]	Write strobes
Data channel APB master (HPM), (Valid when data interface is enabled)			
apb_addr_o	output	[31:0]	Bus configuration address
apb_sel_o	output	1	Bus configuration selection
apb_ena_o	output	1	Bus configuration enable
apb_wr_o	output	1	Bus I/O signal 1: Write 0: Read
apb_rdata_i	input	[31:0]	Read data
apb_wdata_o	output	[31:0]	Write data
apb_rdy_i	input	1	Ready signal
apb_err_i	input	1	Error indication
apb_strb_o	output	[3 :0]	Write strobes
CSR Interface			
cfg_clk_i	input	1	CSR interface clock CSR interface is APB interface, which is used for the data channel when the data channel is enabled
cfg_rstn_i	input	1	Configuration reset
cfg_addr_i	input	[31:0]	Bus configuration address
cfg_sel_i	input	1	Bus configuration selection
cfg_ena_i	input	1	Bus configuration enable
cfg_wr_i	input	1	Bus I/O signal 1: Write 0: Read
cfg_rdata_o	output	[31:0]	Read data
cfg_wdata_i	input	[31:0]	Write data
cfg_rdy_o	output	1	Ready signal
cfg_strb_i	input	[3 :0]	Write strobe signal, which is fixed to 4'hf internally
Debug interface			
debug_o	output	[5:0]	–

Note!

- ^[1] In the table, N is the number of LL GPIOs, you can set via GUI.
- ^[2] In the table, M is the number of NL GPIOs, you can set via GUI.
- ^[3] In the table, L is the number of UARTs, you can set via GUI.
- ^[4] In the table, K is the number of IIC/SMBus, you can set via GUI.
- ^[5] In the table, P is the number of OEMs, you can set via GUI.

3.3 Interface Timing

3.3.1 Configuration Interface Timing

Internal registers are configured via the bus. Figure 3-5 and Figure 3-6 are the write and read timing of the bus configuration. This interface timing is consistent with the one of APB slave. And the interface can be directly connected to the APB master module.

Figure 3-5 Write Timing Diagram of Configuration Interface

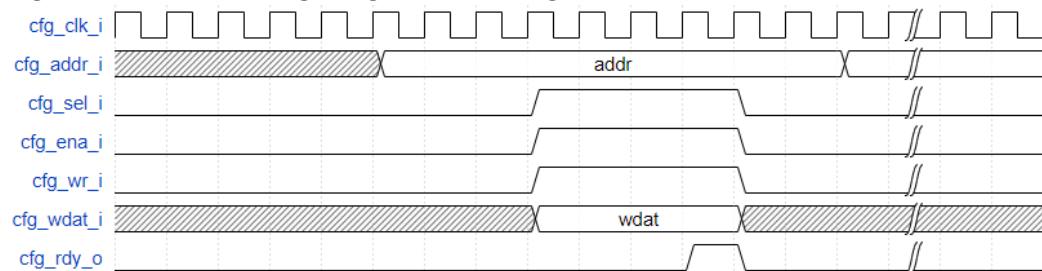
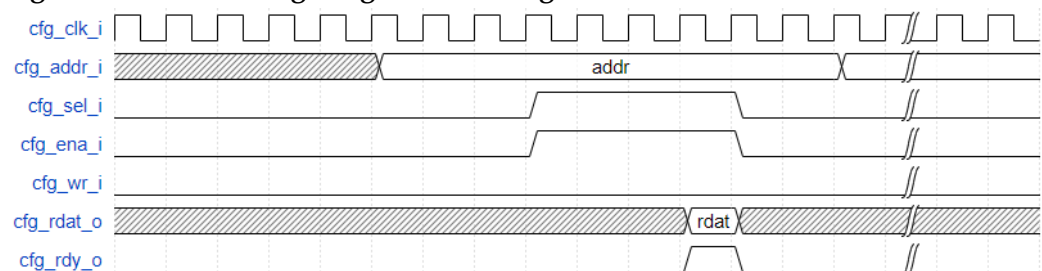


Figure 3-6 Read Timing Diagram of Configuration Interface



3.3.2 Data Interface Timing

This interface is also an APB interface and the timing is the same as configuration interface timing.

3.3.3 GPIO Interface Timing

GPIO interfaces are connected directly.

3.3.4 UART Interface Timing

UART interface timing is standard.

3.3.5 IIC/SMBus Interface Timing

IIC/SMBus bus timing is standard.

3.4 Register Description

Gowin DC-SCM LTPI IP contains LTPI control and status registers (CSR), and is configured through bus interface with address in bytes.

Table 3-2 Gowin DC-SCM LTPI IP Register

Address Offset	Bit	Default	W/R	Description
0x00	31:24	—	—	Reserved
	19:16	—	R	Local LTPI link state 0: Link detection phase 1: Rate negotiation phase 2: Advertisement phase 3: Configuration phase (SCM), Receive phase (HPM) 4: Running status 5-7: Reserved
	15:12	—	R	Remote LTPI link state 0: Link detection phase 1: Rate negotiation phase 2: Advertisement phase 3: Configuration phase (SCM), Receive phase (HPM) 4: Running status 5-7: Reserved
	11:8	—	R	LTPI speed grade (LVDS clock frequency) 0: Base frequency X1 1: Base frequency X2 2: Base frequency X3 3: Base frequency X4 4: Base frequency X6 5: Base frequency X8 6: Base frequency X10 7: Base frequency X12 8: Base frequency X16 9: Base frequency X24 10: Base frequency X32 11: Base frequency X40 12-15: Reserved The base frequency is 25MHz
	7	—	R	DDR Mode 0: SDR Mode 1: DDR Mode
	6	—	—	Reserved
	5	—	R	Link configuration/handshake timeout error
	4	—	R	Link rate negotiation timeout error
	3	—	R	Illegal symbol of comma error

Address Offset	Bit	Default	W/R	Description
	2	–	R	Frame CRC error
	1	–	R	LTPI link loss error
	0	–	R	Link alignment status 0: Link misalignment 1: Link alignment
0x04	31:24	–	–	Reserved
	23:8	GUI Configuration	RW	Local link speed capability, see Table 3-3 for details
	7:4	GUI Configuration	R	Local LTPI Major version Use BCD encoding
	3:0	GUI Configuration	R	Local LTPI Minor version Use BCD encoding
0x08	31:24	–	–	Reserved
	23:8	–	R	See Table 3-4 for remote link speed capability
	7:4	–	R	Remote LTPI Major version Use BCD encoding
	3:0	–	R	Remote LTPI Minor version Use BCD encoding
0x0C	31:16	–	–	Reserved
	15:0	0x0	RW	LTPI local platform ID
0x10	31:16	–	–	Reserved
	15:0	–	R	LTPI remote platform ID
0x14	31:0	GUI Configuration	RW	Local capability advertisement frame content, low 32 bits, see Table 3-4 for details
0x18	31:0	GUI Configuration	RW	Local capability advertisement frame content, high 32 bits, see Table 3-4 for details
0x1C	31:0	–	R	Remote capability advertisement frame content, low 32 bits, as detailed in Table 3-4
0x20	31:0	–	R	Remote capability advertisement frame content, high 32 bits, see Table 3-4 for details
0x24	31:0	SCM: When configuring the GUI, it is default to be the same as the local capability advertisement frame	SCM: RW HPM: R	SCM: LTPI default capability configuration (low 32bits) HPM: Configuration that LTPI receives and takes effect (low 32bits) see Table 3-4 for details
0x28	31:0	SCM: When configuring the GUI, it is default to be the same as the local capability	SCM:RW HPM: R	SCM: LTPI default capability configuration (high 32bits) HPM: Configuration that LTPI receives and

Address Offset	Bit	Default	W/R	Description
		advertisement frame		takes effect (high 32bits) See Table 3-4 for details
0x2C	31:0	0x0	RWC	LTPI link misalignment error counter
0x30	31:0	0x0	RWC	LTPI link loss error counter
0x34	31:0	0x0	RWC	LTPI CRC error counter
0x38	31:0	0x0	RWC	Illegal comma error counter
0x3C	31:0	0x0	RWC	Link speed negotiation timeout error counter
0x40	31:0	0x0	RWC	SCM: Link configuration timeout error counter HPM: Link receiving timeout error counter
0x44	31:24	0x0	RWC	Configure Frame receive counter (HPM Only)
	23:16	0x0	RWC	Speed Frame receive counter
	15:0	0x0	RWC	Detect Frame receive counter
0x48	31:0	0x0	RWC	Advertise Frame receive counter
0x4C	31:24	0x0	RWC	Configure Frame send counter (HPM only)
	23:16	0x0	RWC	Speed Frame send counter
	15:0	0x0	RWC	Detect Frame send counter
0x50	31:0	0x0	RWC	Operational Frame receive counter
0x54	31:0	0x0	RWC	Operational Frame send counter
0x80	31:12	–		Reserved
	10	0x0	RW	Trigger configuration status
	9	0x0	RW	Automatically jump to configuration status 1: Automatically jump 0: Trigger required
	8	0x0	RW	Data channel reset 1: Reset 0: Operating
	7:2	0x0	RW	LTPI IIC channel reset, with each bit indicating each IIC channel 1: Reset 0: Operating
	1	0x0	RW	LTPI link retraining request Retrained on the rising edge
	0	0x0	RW	LTPI link soft reset Soft reset on the rising edge

Note!

R indicates read, W indicates write, WC indicates write clear.

Table 3-3 Speed Description

Byte	Bit Definition							
	7	6	5	4	3	2	1	0
0	Base frequency x12 300 MHz	Base frequency x10 250 MHz	Base frequency x8 200 MHz	Base frequency x6 150 MHz	Base frequency x4 100 MHz	Base frequency x3 75 MHz	Base frequency x2 50 MHz	Base frequency x1 25 MHz
1	0: SDR 1: DDR	Reserved	Reserved	Reserved	Base frequency x40 1 GHz	Base frequency x32 800 MHz	Base frequency x24 50 MHz	Base frequency x16 400 MHz

Table 3-4 Advertisement Frame Capability Description

Byte	Bit Definition							
	7	6	5	4	3	2	1	0
0	Supported Channels							
	Reserved			OEM channel	Data channel	UART channel	IIC channel	GPIO channel
1	GPIO Channel Capabilities							
	Number of NL GPIOs[7:0]							
2	Reserved						Number of NL GPIOs[9:8]	
3	IIC Channel Capabilities							
	Reserved	EchoSupport 1: Support 0: No support	IIC5 En 1: Enable 0: Disable	IIC4 En 1: Enable 0: Disable	IIC3 En 1: Enable 0: Disable	IIC2 En 1: Enable 0: Disable	IIC1 En 1: Enable 0: Disable	IIC0 En 1: Enable 0: Disable
4	Reserved		IIC5Speed 1:400K 0:100K	IIC4Speed 1:400K 0:100K	IIC3Speed 1:400K 0:100K	IIC2Speed 1:400K 0:100K	IIC1Speed 1:400K 0:100K	IIC0Speed 1:400K 0:100K
5	IIC Channel Capabilities							
	Reserved	UART1 En 1: Enable 0: Disable	UART0 En 1: Enable 0: Disable	Flow control	The definition of Max Baud Rate is shown in Table 3-5.			
6	OEM Capabilities							
7								

Table 3-5 Maximum Baud Rate of Serial Port Channels

Baud rate (bps)	Max baud rate encoding
300	0h00
600	0h01
1200	0h02
1800	0h03
2400	0h04
4800	0h05
9600	0h06
19200	0h07
38400	0h08
57600	0h09
115200	0h0A
230400	0h0B
460800	0h0C
576000	0h0D
921600	0h0E
Reserved	0h0F

An example of register setting is as follows:

If the local speed supports X1, X2, X3, X4, X6, DDR; the channel supports GPIO, IIC/SMBus, UART, OEM, DATA, and there are

- 16 LLGPIOs
- 32 NLGPIOs
- 1 IIC with 100K rate supporting Echo
- 32 GPIOs for OEM
- Data channel support
- 1 UART, which does not support flow control while supports a maximum baud rate of 115200bps

The register is set to: 0x04[23:8] = 16h801f, 0x14[31:0] = 32h4100201f, 0x18[31:0] = 32h00002A00.

4 Design Description

4.1 Line Rate

The DC-SCM 2.0 LTPI specification defines 12 LVDS clocks, including X1 (25MHz), X2 (50MHz), X3 (75MHz), X4 (100MHz), X6 (150MHz), X8 (200MHz), X10 (250MHz), X12 (300MHz), X16 (400MHz), X24 (600MHz), X32 (800MHz), and X40 (1000MHz). There are SDR and DDR modes. If it is SDR mode, the line rate is the same as the clock frequency. For example, if you operate in SDR mode with X6 (150MHz), then the line rate is 150Mbps; and if you operate in DDR mode with X6 (150MHz), then the line rate is 300Mbps.

4.2 Clock

4.2.1 System Clock

System clock must be fixed at 60MHz.

4.2.2 Reference Clock

Reference clock must be fixed at 25MHz and the reference clock needs to share the same source with system clock.

4.2.3 Configuration Clock

The core reads and writes registers through configuration interface, and if the data channel is enabled, the clock also acts on the APB interface of the data channel. The clock for this interface is a clock independent of the system clock and reference clock, and there is no dependency between this clock and the system clock or reference clock.

4.2.4 LVDS Clock

There is a dynamically configurable PLL module in the IP core, which generates two clocks, LINKCLK and LINKCLK90, with a phase difference of 90° by default (this phase difference can be adjusted by the registers). Based on the results of rate negotiation, dynamic LVDS frequency configuration is performed to drive the generation of TX LVDS CLK and TX LVDS Data signals.

4.2.5 RX/TX DIV5 Clock

The IP core internally incorporates the serial-to-parallel conversion primitive ISER10/OSER10, with the clock for parallel data being divided by 5 from LVDS CLK. In the TX, TXDIV5 is divided by 5 from the LINKCLK with the CLKDIV primitive. In the RX, RXDIV5 is divided by 5 from the received LVDS CLK with the CLKDIV primitive.

4.2.6 Clock Structure

Figure 4-1 Typical Clock Scheme

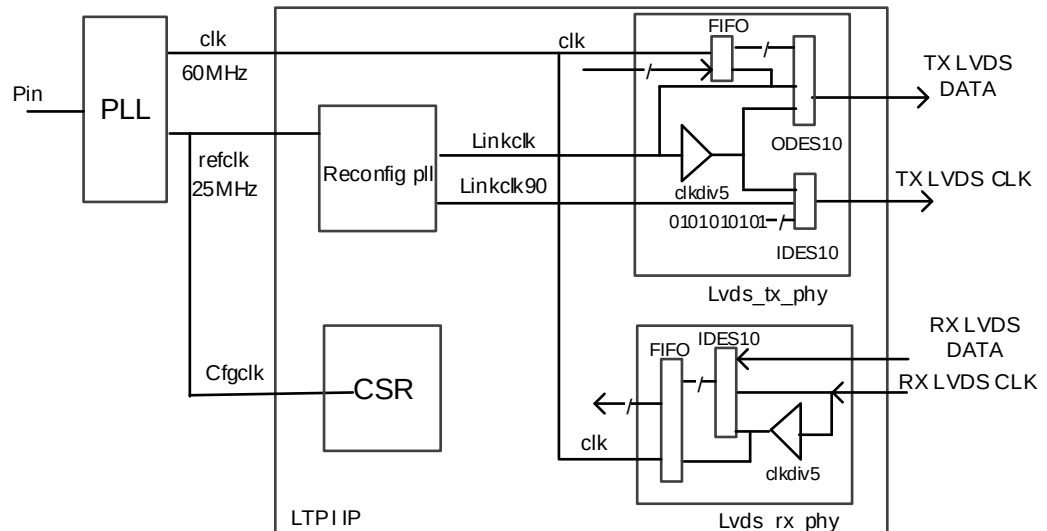


Figure 4-1 shows a clocking scheme based on the same clock source, generating the system clock (clk) at 60MHz, the reference clock (refclk) at 25MHz, and the configuration clock (cfgclk) can be the same as the reference clock.

4.3 External Channel Interface Processing

LTPI uses the following universal methods to acquire and transmit various LTPI channels:

- **Sampling:** The I/O state is sampled by LTPI, which is tunneled in LTPI frames.
- **Event/State Detection:** A set of events/states is defined for the channel based on the interface status and tunneled in LTPI frames.
- **Random Access Read/Write Requests:** The external interface triggers a random access memory-mapped channels.

The interfaces supported by LTPI can also be classified according to the RX and TX synchronization of directions of a given interface, i.e. whether the tunneling of the interface requires synchronization between the TX and RX directions:

- **Asynchronous:** For a given channel and interface/link, LTPI TX and RX paths are independent.
- **Synchronization:** For a given channel and interface/link, LTPI TX and

RX paths need to be synchronized to allow the interface to operate correctly after tunneling.

Table 4-1 External Channel Interface Processing

Channel Type	Acquisition Mode	TX and RX Synchronization Mode	Channel Feature
GPIO	Sample	Asynchronization	Acquire signal levels directly. For low-latency GPIOs, each LTPI frame needs to be updated. For normal-latency GPIOs, the frame number can be larger, but since LTPI frames have given a limited locations for normal-latency GPIOs, they may be updated every few frames, and LTPI frames have a frame ID flag to identify the location of the normal-latency GPIOs.
UART			Acquire signal levels directly. UART signals are oversampled. Each LTPI frame contains multiple sample values.
IIC/SMBus	Transaction/State Detection	Synchronization	IIC/SMBus state is encoded into LTPI transactions and transmitted through LTPI, as IIC/SMBus requires bidirectional interaction. After sending LTPI transactions, it is necessary to wait for the response from the LTPI interface on the other end. Therefore, the clock stretching feature of IIC/SMBus is used to wait for synchronization completion.
Data Bus	Random access		Data bus transmissions such as data read and write encoded as LTPI transactions. The bus operation completion indicator is used to synchronize access to the data bus.

4.3.1 GPIO Channel

For GPIO channels, the GPIO logic states (high and low) are directly encoded on the LTPI frame bits. The Low-Latency (LL) GPIO is refreshed in each LTPI frame and has a dedicated bit for each GPIO in each frame sent. Refresh Normal-Latency (NL) GPIO every N frames, where N is defined as:

$$N = \left(\frac{\text{Total Number of NL GPIOs tunneled over LTPI}}{\text{Number of NL GPIO bits per LTPI IO Frame}} \right)$$

N = Number of GPIOs to transmit/Number of NL GPIO bits per IO frame

In order to decode the NL GPIO tunneled in a given frame, the NL GPIO index from the LTPI frame is decoded with a frame counter incremented in each frame. The encoding of the LTPI GPIO channels in LTPI frames is shown in the following table, where LL and NL GPIO are defined on 1 byte, respectively. For LL GPIO, each bit in a byte is defined as a GPIO with a number ranging from 0 to 7. For NL GPIO, in order to decode the GPIO with a number ranging from 0 to 7 for each bit, an offset X is added, where X is defined as:

$$X = (\text{Frame Counter} \% N) \\ * (\text{Number of NL GPIO bits per LTPI Frame})$$

Table 4-2 GPIO Field Configuration

Name	Bit Definition							
	7	6	5	4	3	2	1	0
Low-latency GPIO (LLGPIO)	LL GPIO7 0: Low 1: High	LL GPIO6 0: Low 1: High	LL GPIO5 0: Low 1: High	LL GPIO4 0: Low 1: High	LL GPIO3 0: Low 1: High	LL GPIO2 0: Low 1: High	LL GPIO1 0: Low 1: High	LL GPIO0 0: Low 1: High
Normal-latency GPIO (NLGPIO)	NL GPIO [X+7] 0: Low 1: High	NL GPIO [X+6] 0: Low 1: High	NL GPIO [X+5] 0: Low 1: High	NL GPIO [X+4] 0: Low 1: High	NL GPIO [X+3] 0: Low 1: High	NL GPIO [X+2] 0: Low 1: High	NL GPIO [X+1] 0: Low 1: High	NL GPIO [X] 0: Low 1: High

The default I/O frame allocates 2 bytes for the LL GPIO and 2 bytes for the NL GPIO. This means that 16 NL GPIOs can be sent in each IO frame and N frames are required to transmit all NL GPIOs. If an implementation needs to define more LL or NL GPIOs in a single IO frame, the OEM field can be used to extend the default I/O frame.

The total number of NL GPIOs is defined in the capability frame of “NL GPIO Number” field. Capabilities are exchanged during the LTPI training phase. In this way, LTPI in SCM and LTPI in HPM can correctly decode the index of NL GPIO in a given frame based on the frame counter and N parameter defined above.

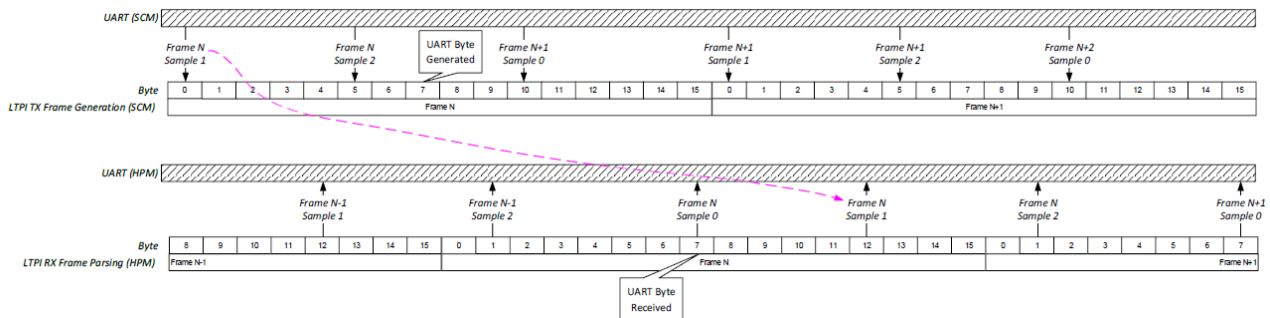
4.3.2 UART Channel

The UART channel operates like the LL GPIO channel. The difference is that the sampling frequency for UART transmit data (TXD) and receive data (RXD) lines is three times the sampling frequency of LL GPIO.

How the UART sampled data goes from the TX side to the RX side via LTPI frames and the UART data is recovered at the receiver side is shown in Figure 4-2. As shown in this example, UART samples need to be ready when byte No. 7 is generated and transmitted through LTPI. This means that before this, it is necessary to collect 3 samples, assuming that the samples are evenly distributed over time. For frame N, sample 0 is actually acquired when byte 10 of the previous frame (N-1) is transmitted. Sample 1 is acquired when byte 0 is transmitted through LTPI and sample 2 is acquired when byte 7 with UART sample is generated and

transmitted. In this way, the samples are evenly distributed over time between frames. On the receiver side, it is necessary to maintain the same distribution, when the received samples are used to generate signals on the UART interface. When byte 7 of frame N is received, sample 0 is used to generate a signal on the UART interface; When byte 12 is received, the next sample 1 is generated from frame N. The last sample is generated on byte 1 of the next frame (N+1).

Figure 4-2 UART Sampling and Recovery Diagram



The UART channel, along with RXD and TXD signals, supports flow control signal tunneling supported by RTS and CTS. The flow control signal is not oversampled, but is similar to a low-latency GPIO encoded in each LTPI frame. Table 4-3 shows the encoding of the UART channels. Each channel is encoded on 4 bits in TX and 4 bits in RX direction. These encodings are tunneled separately through the LTPI interface. In this example, considering the direction from the SCM, the encoding of UART0/1 TX is applicable to LTPI frames tunneled from SCM to HPM. The encoding of UART0/1 RX is applicable to LTPI frames tunneled from HPM to SCM.

Table 4-3 UART Field Configuration

Name	Bit Definition							
	7	6	5	4	3	2	1	0
UART0/1 TX encoding	UART1				UART0			
	RTS	TXD[2]	TXD[1]	TXD[0]	RTS	TXD[2]	TXD[1]	TXD[0]
	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High
UART0/1 RX encoding	CTS	RXD[2]	RXD[1]	RXD[0]	CTS	RXD[2]	RXD[1]	RXD[0]
	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High	0: Low 1: High

4.3.3 IIC/SMBus Channel

The IIC/SMBus SCL and SDA signals are bidirectional and can be actively driven by the controller or target device during an IIC/SMBus event. Therefore, SCL and SDA signals cannot be tunneled directly through LTPI as GPIO or UART signals. In LTPI, the IIC/SMBus channel implements an event-based model in which encoded IIC/SMBus events/states are captured on one side of the LTPI and recovered on the other. When an IIC/SMBus event is generated by the controller (for example, START condition), the controller's local SCL line is stretched by the LTPI IP until a remote event is received. When a remote event is received, it is recovered on the controller local bus and the SCL is released from the stretching state.

Table 4-4 IIC/SMBus Channel Transaction Definition

Bus event	Direction	Description	Transaction Encoding (4bit)
Idle	Bidirectional	Idle status	4b0000
Start	Controller to Target	The controller detects a start event	4b0001
Start Received	Target to Controller	Start event recovers at the target	4b0010
Stop	Controller to Target	The controller detects a stop event	4b0011
Stop Received	Target to Controller	Stop event recovers at the target	4b0100
Data Received	Bidirectional	SDA value recover at the remote end	4b0101
Data0	Bidirectional	Send SDA value 0	4b0110
Data1	Bidirectional	Send SDA value 1	4b0111
Start Echo	Target to Controller	Feedback transaction that the start event is correctly received by the target	4b1000
Stop Echo	Target to Controller	Feedback transaction that the stop event is correctly received by the target	4b1001
Data0 Echo	Bidirectional	The SDA TX value of 0 is received correctly	4b1010
Data1 Echo	Bidirectional	The SDA TX value of 1 is received correctly	4b1011
Data Received Echo	Bidirectional	The Data Received event is received correctly. It is used to distinguish consecutive Data 0/1 events during the byte transmission process.	4b1100
Reserved	Reserved	Reserved	4b1101 - 4b1111

Each IIC/SMBus bus is equivalent to 4-bit valid payload data, as shown in Table 4-4. In the example in Table 4-5, for the 6 IIC/SMBus buses, 3 bytes of valid payload are allocated to the corresponding IIC/SMBus channels for each frame.

Table 4-5 IIC/SMBus Field Configuration

IIC/SMBus Payload Byte	Bit Definition							
	7	6	5	4	3	2	1	0
0	IIC/SMBus BUS 1				IIC/SMBus BUS 0			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
1	IIC/SMBus BUS 3				IIC/SMBus BUS 2			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]
2	IIC/SMBus BUS 5				IIC/SMBus BUS 4			
	Event[3]	Event[2]	Event[1]	Event[0]	Event[3]	Event[2]	Event[1]	Event[0]

Figure 4-3 IIC/SMBus Bus Transaction Exchange

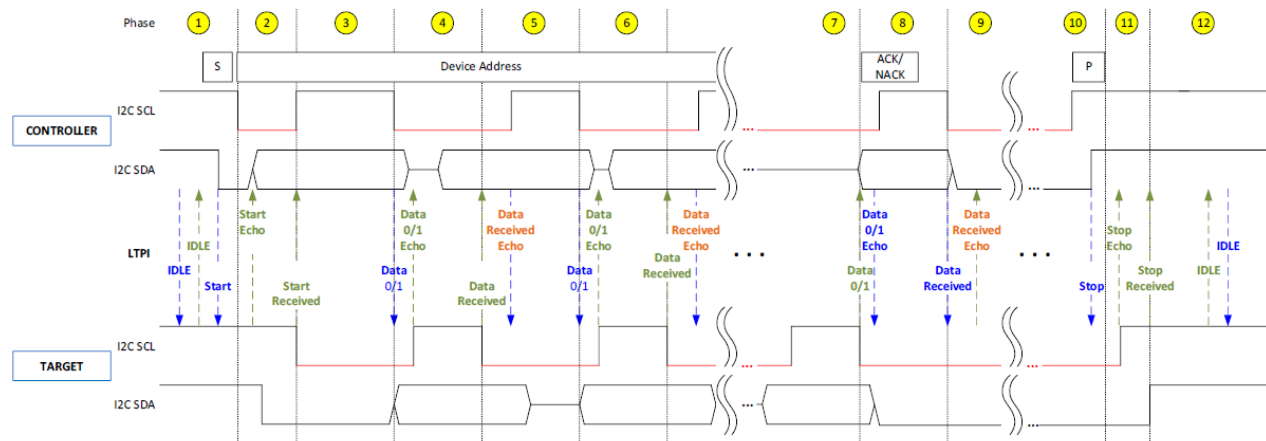


Figure 4-3 shows the IIC/SMBus bus event exchange. The event is initiated by the controller. Once an event request is detected in LTPI, the communicator sends encoded event echo back to the requestor via LTPI to confirm reception, and when the event is successful, it will send an event acknowledgment data packet. When either party waits for feedback from the communicating party, a clock stretching occurs on the local SCL bus to prevent new transactions from occurring while the current transaction is in progress. Encoded packets continue to be transmitted until a new event is generated. The bold line in the SCL interface indicates the clock stretching that is completed on the SCL bus.

Table 4-6 IIC/SMBus Transaction Flow

No.	Controller	Target	IIC/SMBus event	LTPI Event Direction
1	The IP detects the SDA falling edge and the SCL high level and starts sending the Start event to the LTPI interface; The IP triggers the SCL stretching on the local SCL and waits for START Received.	Bus idle status	Start	Controller to Target
2	Clock Stretching	Receive Start events from LTPI, send Start Echo events back for feedback, and generate Start Condition timing on local SCL and SDA buses; Send Start Received events to LTPI; Trigger SCL stretching and wait for next event.	Start Echo Start Received	Target to Controller
3	Stop Clock stretching on local bus after receiving Start Received events from LTPI; Controller starts sending data bits; IP detects SCL rising edge and SDA data for local bus; IP detects SCL falling edge, and sends a Data0/1 transaction to LTPI; IP triggers SCL stretching on local SCL and waits for a Data Received event to be received.	Clock Stretching	Data0/1	Controller to Target
4	Clock Stretching	Receive Data0/1 events from LTPI, send Data Echo events back for feedback, and release SCL and send data 0/1 on local SCL and SDA buses; SCL is pulled down after SCL high duration time; Send Data Received events to LTPI; Trigger SCL stretching and wait for next event.	Data Echo Data Received	Target to Controller
5	Stop Clock stretching on local bus after receiving Data Received events from LTPI; Send Data Received Echo events to LTPI interface and controller start sending data bits; IP detects SCL rising edge and SDA data for local bus; IP detects SCL falling edge, and sends a Data0/1 transaction to LTPI; IP triggers	Clock Stretching	Data Received Echo Data 0/1	Controller to Target

No.	Controller	Target	IIC/SMBus event	LTPI Event Direction
	SCL stretching on local SCL and waits for a Data Received event to be received.			
6	Clock Stretching	Data Received Echo event is received from LTPI first, indicating that the previous bit was sent. Next receive a new Data 0/1 event; Send Data Echo event for feedback, release SCL and send Data 0/1 on local SCL and SDA buses; SCL is pulled down after SCL high duration time; Send Data Received event to LTPI; Trigger SCL stretching and wait next event.	Data Echo Data Received	
...	...	...	...	...
The change in the direction of the data transaction i.e., the ACK/NACK in this example, is the same for the data bits in the read transaction				
...	...	...	...	...
7	Clock Stretching	Target sets the data bit on SDA; IP detects SCL rising edge and SDA data on the local bus; IP detects SCL falling edge and sends Data0/1 events to LTPI; Target triggers SCL stretching and waits for next event.	Data 0/1	Target to Controller
8	Receive Data0/1 events from LTPI, send Data Echo events back, release SCL and send Data 0/1 on local SCL and SDA buses; SCL is pulled down after SCL high duration time; Send Data Received event to LTPI; Trigger SCL stretching and wait for next event.	Clock Stretching	Data Echo Data Received	Controller to Target
9	Clock Stretching	Receive Data Received events from LTPI; Start sending Data Received Echo event.	Data Received Echo	Target to Controller
...	...	...	...	...
10	When the SCL on the local bus is high (Stop condition on the local bus), IP detects the SDA rising edge. IP sends STOP events to	Clock Stretching	Stop	Controller to Target

No.	Controller	Target	IIC/SMBus event	LTPI Event Direction
	LTPI.			
11	Wait for the Stop Echo and Stop Received events; IP waits for the new IIC start condition, and if a new START condition is detected, it should be deferred (using clock stretching) until a Stop Received event for the previous STOP condition is received.	Receive Stop events; Send Stop Echo events and generate Start Condition timing on local SCL and SDA buses; Send Stop Received via LTPI interface; Trigger bus idle status.	Stop Echo Stop Received	Target to Controller
12	Bus Idle Status	Bus Idle Status	Idle	N/A

4.3.4 OEM Channel

The OEM channel is similar to LLGPIO, with the difference that the OEM channel will update during all non-data channel frame transmissions, whereas LLGPIO updates even during data channel frame transmissions.

4.3.5 Data Channel

Data channels allow for tunneling standardized read and write requests to addressable memory space. The DC-SCM LTPI soft IP supports the APB interface of the data channel and can access on demand when enabled. In LTPI, data channels implement an event-based model in which encoded data events/states are captured on one side of LTPI and recovered on the other.

Table 4-7 Data Channel Transaction Definition

Instruction	Parameter	Description	8-bit Event Encoding
Read request	Address	Read request instruction for the specified address	0x00
Write request	Address and data	Write request instruction for the specified address and data	0x01
Read done	Address and data	Read done instruction	0x02
Write done	Address	Write done instruction	0x03
CRC error	None	Data frame CRC error received and the operation can not be done	0x04
Reserved	Reserved	Reserved	0x05~0xFF

Based on the instruction request, each data channel byte is mapped to each payload byte in the data frame.

Table 4-8 Data Channel Field Configuration

Name	Bit encoding							
	7	6	5	4	3	2	1	0
Read operation	Command = 8'h00							
	Address byte 3							
	Address byte 2							
	Address byte 1							
	Address byte 0							
	Reserved				Byte 3 Enable	Byte 2 Enable	Byte 1 Enable	Byte 0 Enable
Read operation done	Command = 8'h02							
	Address byte 3							
	Address byte 2							
	Address byte 1							
	Address byte 0							
	Read operation status 0: Success, 1: Illegal access, 2-F reserved				Byte 3 Enable	Byte 2 Enable	Byte 1 Enable	Byte 0 Enable
	Data Byte 3							
	Data Byte 2							
	Data Byte 1							
	Data Byte 0							
Write Operation	Command = 8'h01							
	Address byte 3							
	Address byte 2							
	Address byte 1							
	Address byte 0							
	Reserved				Byte 3 Enable	Byte 2 Enable	Byte 1 Enable	Byte 0 Enable
	Data Byte 3							
	Data Byte 2							
	Data Byte 1							
	Data Byte 0							
Write operation done	Command = 8'h01							
	Address byte 3							
	Address byte 2							
	Address byte 1							
	Address byte 0							
	Write operation status 0: Success, 1: Illegal access, 2-F reserved				Reserved			

4.4 LTPI Frame Format

4.4.1 Frame Format

The frame format in the IP is all consistent with those specified in the DC-SCM 2.0 LTPI protocol, and all types of frames are fixed-length, 16 bytes in total, as shown in Table 4-9.

Table 4-9 Frame Format Description

Byte Count	Description
0	Frame start, K-character
1	Frame type
2-14	Frame data
15	CRC8 checksum, frame end

4.4.2 Specific Frame Definition

1. Link training phase

There are two subframe types for this phase, as shown in Table 4-10.

Table 4-10 Link Training Phase Frame Type

Frame Type	Comma Symbol	Frame Type Value
Link Detect	K28.5	0x00
Link Speed		0x01

a. Link Detect Frame

First, the Link Detect Frame is sent from both sides, as defined in Table 4-11.

Table 4-11 Link Detect Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	LTPI Major version				LTPI Minor version				BCD Encoding [Major].[Minor]
3	Speed capability								See Table 3-3 for format
4									
5-14	Reserved								Reserved
15	CRC								CRC8 checksum

b. Link Speed Frame

The Link Speed Frame format is identical to the Link Detect Frame format except that Speed Select is the negotiated rate in the Link Speed Frame as shown in Table 4-12.

Table 4-12 Link Speed Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	LTPI Major version				LTPI Minor version				BCD Encoding [Major].[Minor]
3	Speed after negotiation								See Table 3-3 for format
4									
5-14	Reserved								Reserved
15	CRC								CRC8 checksum

2. Link configuration phase

There are three subframe types for this phase, as shown in Table 4-13.

Table 4-13 Link Configuration Phase Frame Type

Frame Type	Comma Symbol	Frame Type Value
Advertise	K28.6	0x00
Configure		0x01
Accept		0x02

a. LTPI Advertise Frame

After rate negotiation is complete, both SCM and HPM send Advertise frames to advertise the ability to each other, as defined in Table 4-14.

Table 4-14 Advertise Frame Definition

Intra-frame Byte	Bit content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	Platform type								—
3									
4	Capability type 0x00 Default capability type 0x01-0x80 Reserved 0x81-0xFF OEM custom								The capability type allows for custom LTPI capabilities
5	LTPI capability								See Table 3-4 for format
6									
7									
8									
9									

Intra-frame Byte	Bit content								Description
	7	6	5	4	3	2	1	0	
10									
11									
12									
13-14	Reserved								Reserved
15	CRC								CRC8 checksum

After advertising the capabilities to each other, SCM sends Configure frames. Configure frames are in the same format as Advertise frames, except that SCM places either the capabilities preset by SCM or the capabilities set in real-time by the BMC in the Request LTPI Capability field.

b. LTPI Configure Frame

Table 4-15 Configure Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	Platform type								–
3									
4	Capability type 0x00 Default capability type 0x01-0x80 Reserved 0x81-0xFF OEM custom								The capability type allows for custom LTPI capabilities
5	Request for LTPI capability								See Table 3-4 for format
6									
7									
8									
9									
10									
11									
12									
13-14	Reserved								Reserved
15	CRC								CRC8 checksum

c. LTPI Accept Frame

When receiving Configure frames, HPM compares with the local capacity and sends Accept frames if the local capacity contains the capability described in the Configure frame. The Accept frame is in the same format as the Configure frame. Some of the contents of the Accept LTPI Capability and Request LTPI Capability are the same.

Table 4-16 Accept Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	Platform type								–
3									
4	Capability type 0x00 Default capability type 0x01-0x80 Reserved 0x81-0xFF OEM custom								The capability type allows for custom LTPI capabilities
5	Accept LTPI capability								See Table 3-4 for format
6									
7									
8									
9									
10									
11									
12									
13-14	Reserved								Reserved
15	CRC								CRC8 checksum

After receiving the Accept frame, SCM begins to operate and sends the Operation frame, while HPM enters the operation phase when it receives the Operation frame from the SCM.

3. Operation Phase

When both SCM and HPM are in Operation state, start sending operation frames.

There are three types of operation frames, the default I/O frame, the default data frame, and the OEM custom frame, as shown in Table 4-17. The IP only supports default I/O frames and default data frames currently.

Table 4-17 Operation Phase Frame Type

Frame Type	Comma Symbol	Frame Type Value
Default I/O	K28.7	0x00
Default Data		0x01
Reserved		0x02 - 0x0F
OEM Defined		0x10 - 0xFF

When there is no enabled Data channel, there is only Default I/O Frame and LTPI Default I/O Frame.

a. LTPI Default I/O Frame

This frame transmits information for channels such as GPIO/UART/IIC/OEM, as defined in Table 4-18.

Table 4-18 Default I/O Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	Frame counter								The frame counter is used to decode the location of NL GPIO
3	Low-latency GPIO0								See Table 4-2 for definitions
4	Low-latency GPIO1								See Table 4-2 for definitions
5	Normal-latency GPIO0								See Table 4-2 for definitions
6	Normal-latency GPIO1								See Table 4-2 for definitions
7	UART0 and UART1								See Table 4-3 for definitions
8	IIC0 and IIC1								See Table 4-5 for definitions
9	IIC2 and IIC3								See Table 4-5 for definitions
10	IIC4 and IIC5								See Table 4-5 for definitions
11-14	OEM reserved								OEM reserved
15	CRC								CRC8 checksum

b. LTPI Default Data Frame

This frame is used to transmit bus data, as defined in Table 4-19.

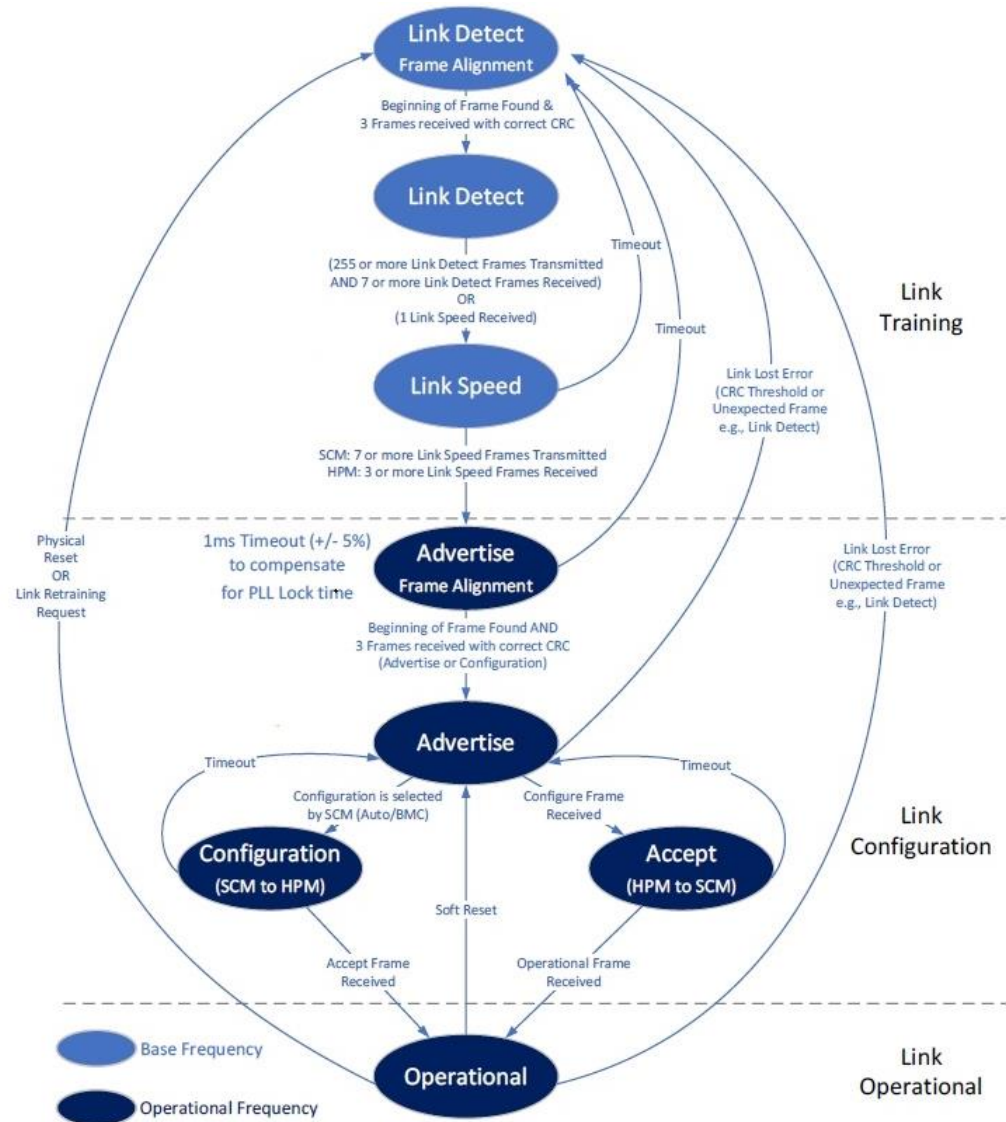
Table 4-19 Default Data Frame Definition

Intra-frame Byte	Bit Content								Description
	7	6	5	4	3	2	1	0	
0	Comma symbol								Defined according to the table above
1	Frame type								Defined according to the table above
2	Low-latency GPIO0								See Table 4-2 for definitions
3	Low-latency GPIO1								See Table 4-2 for definitions
4	Tag								-
5-14	Data channel payload								See Table 4-8 for definitions
15	CRC								CRC8 checksum

4.5 Link Initialization & Operation Flow

4.5.1 State Diagram

Figure 4-4 LTPI Link State Diagram



LTPI Link Training and Switch to Operation Mode are defined as a state machine that can be divided into three major phases:

- Link Training: The initial state of link initialization
- Link Configuration: Configuration of LTPI features
 - Exchange LTPI feature capabilities
 - Enable the selected operation configuration of LTPI
 - Switch to Operation Mode
- Link Operation: The link is available for most of the LTPI lifecycle

The operation of the entire link follows the process of the state machine.

4.5.2 Link Training

The Link Training phase starts after the SCM and HPM come out of reset. It is executed on Base Frequency of 25MHz in Single Data Rate (SDR) mode. The main purposes of the Link Training phase include:

- Achieve DC-balance condition on the link
- Exchange Link Speed capabilities
- Switch to Link Operation Frequency

The Link Training phase is based on 2 states:

- Link detect state (Link Detect)
- Link speed negotiation state (Link Speed)

In the Link Detect, both SCM and HPM start sending Link Detection frames. Frames are transmitted in two directions. The main purpose is to establish link DC balance and indicate the supported operation frequency. This phase can be divided into 2 parts:

- Link Detect-Frame Alignment: In this part, the SerDes logic attempts to lock the beginning of the frame by identifying the comma symbol and processing the frame to verify the CRC. In order to move to the next part, at least 3 frames with the correct CRC need to be received.
- Link Detect: This is the main part of Link Detect where frames are used to interpret the operation speed capabilities of the other side. In this part, TX and RX counters are being used. Due to the timing differences between SCM and HPM in achieving DC-balance finding beginning of the frame, it is expected that SCM and HPM TX/RX counters will be misaligned.

Both SCM and HPM shall:

- Send at least 255 Link Detect Frames on TX Link
- Receive at least 7 consecutive frames with correct CRC checksum on RX Link

However, if one side moves to the next phase faster and starts sending Link Speed frame, the “slower” side shall move to next state immediately.

In the link speed negotiation state, both SCM and HPM start sending Link Speed frames. SCM shall pick up the highest common target frequency based on the Link Detect capabilities received and its own capabilities. In some cases, like system integration, system debugging or recovery mode, it might be needed to lower the operation frequency below maximum capabilities. In order to do that, LTPI can expose a setting to BMC or other controller on the SCM to allow the modification of SCM LTPI Link Detect capabilities.

The following conditions shall be met to move to next phase:

- SCM: Send at least 7 Link Speed frames on the TX link
- HPM: Receive at least 3 Link Speed frames

4.5.3 Link Configuration

The Link Configuration starts after Link Training phase is completed. It is executed on operation frequency selected during Link Speed phase. The main purposes of Link Configuration phase include:

- Switch to operational link frequency
- Exchange LTPI capability
- Enable the selected operation configuration of LTPI
- Switch to Operation Mode

The Link Training phase is based on 3 states:

- Advertise
- Configure
- Accept

The Advertise state is the first state when link switches from base frequency to the operation frequency. In the Advertise state, both SCM and HPM start sending the Advertise frame. Frames are transmitted in two directions. After switching to operation frequency, the RX SERDES needs to re-adjust to the beginning of the frame similarly as in Link Detect phase. The phase can also be divided into 2 parts:

- Advertise-Frame Alignment: In this part the SerDes logic is trying to lock to the beginning of the frame by identifying the comma symbol and processing the frame to verify the CRC. In order to move to the next part, at least 3 frames with the correct CRC need to be received.
- Advertise: This is the main part of Advertise where frames are 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 operation frequency. If SCM completes 1ms transmission requirement first and automatically starts sending Configure Frame, HPM shall immediately switch to Accept without completing the 1ms transmission requirement. In Advertise state, SCM and HPM keep sending Advertise frames. This state does not have a timeout for proceeding to the next state, which is to allow BMC to read the Advertise information and select the LTPI configuration. BMC can use the CSR registers to verify that LTPI Advertise state is active, read advertise information, and write the target configuration of the LTPI.

For SCM and HPM, the next states are different. SCM enters Configure state and HPM Accept state. To move to the next state, SCM needs to start sending Configure frames:

- Automatically based on SCM LTPI configuration after the Advertise state is entered correctly
- On request from BMC, the following configuration is applied by the BMC

If 3 consecutive Advertise frames are lost in Advertise state, then the

LTPI shall return to Link Detect state.

SCM enters Configure state and HPM enters Accept state; Configure frames are being transmitted by SCM only and Accept Frames are being transmitted by HPM only. In this state, SCM starts sending Configure frames and waits for HPM to send the Accept frames back.

To move to the next state:

- The SCM starts sending Configure frames to HPM
- When receiving at least 1 Configure frame and matching HPM capabilities, HPM sends the Accept frame back to SCM with the same capabilities as requested by SCM.
- When receiving Accept frame and matching Configure Frame capabilities, SCM moves to next state by sending the first Operation frame on the link.

If SCM do not receive the matching Accept frame after sending 32 Configure frames, it moves back to Advertise state.

4.5.4 Link Operating

The Operation Mode is the main operating condition for the majority of LTPI lifetime. In Operation Mode, both SCM and HPM constantly send I/O frames in both directions.

When a data channel is used, the data frames will be generated on demand when is triggered by the following method: When Data Channel frame is generated, it will interleave the I/O frame stream.

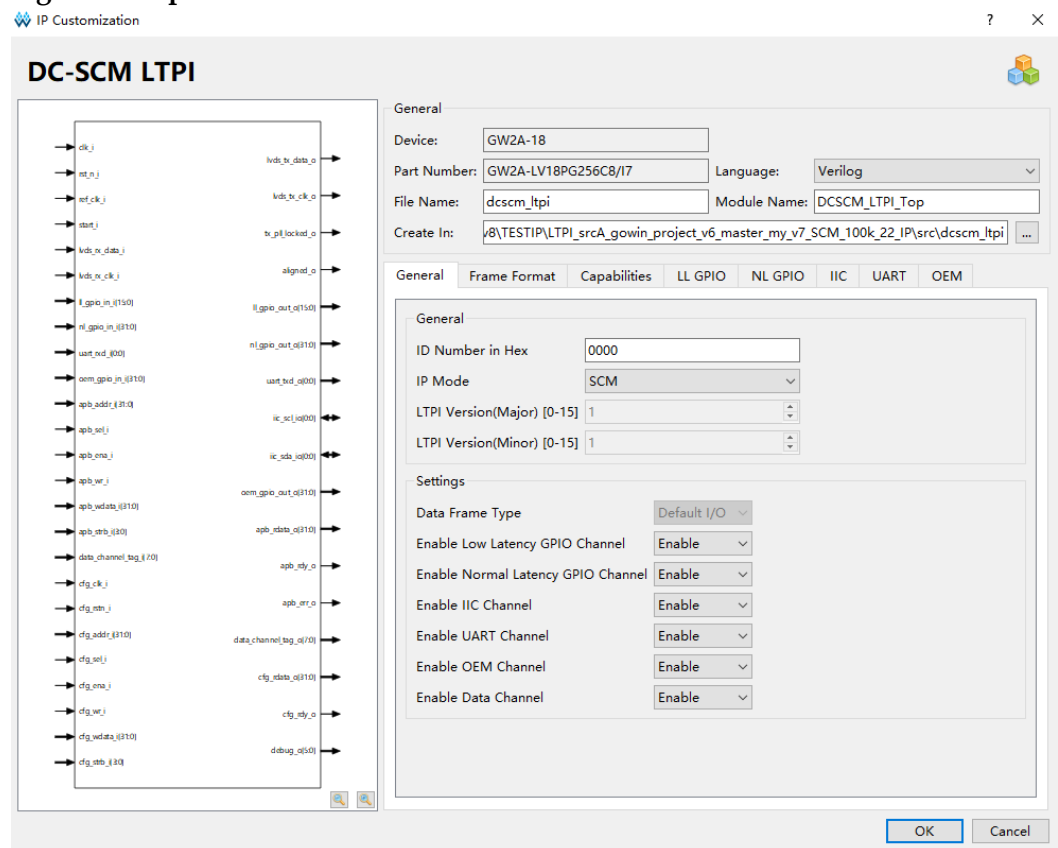
The SCM and HPM remain in the operation state until following conditions occur:

- Soft Reset is triggered: Switch to Advertise state
- Hard Reset is triggered (FPGA Reset, Power Lost): Start in Link Detect state
- Link is lost after 7 consecutive frames are lost: Start in Link Detect state

5 Interface Configuration

Find Soft IP Core in the IP library and click DC-SCM LTPI IP in the Interface and Interconnect field, as shown in Figure 5-1.

Figure 5-1 Open DC-SCM LTPI IP



Gowin DC- SCM LTPI IP configuration interface is shown in Figure 5-2.

Figure 5-2 Gowin DC-SCM LTPI IP Configuration Interface 1

The screenshot shows the 'General' tab of the Gowin DC-SCM LTPI IP Configuration Interface. The interface is divided into two main sections: 'General' and 'Settings'. In the 'General' section, there are four fields: 'ID Number in Hex' with the value '0000', 'IP Mode' set to 'SCM', 'LTPI Version(Major) [0-15]' set to '1', and 'LTPI Version(Minor) [0-15]' set to '1'. The 'Settings' section contains a 'Data Frame Type' dropdown set to 'Default I/O', and seven checkboxes for enabling different channels: 'Enable Low Latency GPIO Channel', 'Enable Normal Latency GPIO Channel', 'Enable IIC Channel', 'Enable UART Channel', 'Enable OEM Channel', and 'Enable Data Channel'. All these checkboxes are currently set to 'Enable'.

- ID Number in Hex: Enter a 16-bit width hexadecimal number to fill the Platform ID field
- IP Mode: Select whether the local is SCM or HPM
- LTPI Version (Major): The version number, fixed to 1, cannot be edited
- LTPI Version (Minor): The version number, fixed to 1, cannot be edited
- Data Frame Type: Custom type is not supported temporarily, only default type is supported
- Enable Low Latency GPIO Channel
- Enable Normal Latency GPIO Channel
- Enable IIC Channel: Enable IIC/SMBus channel
- Enable UART Channel
- Enable OEM Channel
- Enable Data Channel

Figure 5-3 Gowin DC-SCM LTPI IP Configuration Interface 2

Field	Value
Frame Offset Byte 0	Start of Frame
Frame Offset Byte 1	Frame Sub-Type
Frame Data Sequence 0	NL Frame Counter
Frame Data Sequence 1	LL GPIO
Frame Data Sequence 2	NL GPIO
Frame Data Sequence 3	UART
Frame Data Sequence 4	IIC
Frame Data Sequence 5	OEM
Frame Offset Byte 15	CRC

Figure 5-3 mainly shows the structure of the Data I/O frame, since there is only default mode currently, the locations of the various channels are fixed, the entire page is not editable.

Figure 5-4 Gowin DC-SCM LTPI IP Configuration Interface 3

Speed Capability

☒ Dual-Data Rate(DDR)	
☒ X1(25MHz)	☐ X10(250MHz)
☒ X2(50MHz)	☐ X12(300MHz)
☒ X3(75MHz)	☐ X16(400MHz)
☒ X4(100MHz)	☐ X24(600MHz)
☒ X6(150MHz)	☐ X32(800MHz)
☐ X8(200MHz)	☐ X40(1000MHz)

Feature Capability

☒ Automatically move to Configuration State

- **Speed Capability:** The IP supports X1, X2, X3, X4, X6, and DDR modes currently and allows you to choose the speed and mode that IP needs to support based on device capabilities and actual applications.
- **Automatically move to Configuration State:** Only SCM can be edited. When the option is checked, the Configure state is automatically entered after the Advertise state ends. Otherwise, a register trigger is required to enter the Configure state.

Figure 5-5 Gowin DC-SCM LTPI IP Configuration Interface 4

General Frame Format Capabilities LL GPIO NL GPIO IIC UART OEM

Setting

LL GPIO Data Width(1-16) 16

- LL GPIO Data Width: Low-latency GPIO count, up to 16

Figure 5-6 Gowin DC-SCM LTPI IP Configuration Interface 5

General Frame Format Capabilities LL GPIO NL GPIO IIC UART OEM

Setting

NL GPIO Data Width(1-1023) 32

- NL GPIO Data Width: Normal-latency GPIO count, up to 1023

Figure 5-7 Gowin DC-SCM LTPI IP Configuration Interface 6

General Frame Format Capabilities LL GPIO NL GPIO IIC UART OEM

Setting

Number of IIC bus interface(1-6) 1

☒ IIC Echo Support

IIC BUS 0 Mode Controller

☐ Enable Fast Mode for BUS 0

- Number of IIC bus interface: IIC bus interface count, up to 6
- IIC Echo Support: Default support
- IIC BUS 0 Mode: When IP mode is SCM, the default is Controller; When IP mode is HPM, the default is Target, user configurable
- Enable Fast Mode for BUS: IIC/SMBus supports 400KHz when checked, 100KHz when unchecked

Figure 5-8 Gowin DC-SCM LTPI IP Configuration Interface 7

The screenshot shows the 'UART' configuration tab. It contains three settings: 'Baud Rate' set to 115200, 'Enable Flow Control' set to Disable, and 'Number of UART bus interface(1-2)' set to 1.

Setting	Value
Baud Rate	115200
Enable Flow Control	Disable
Number of UART bus interface(1-2)	1

- Baud Rate: Baud Rate setting
- Enable Flow Control: Flow control enable
- Number of UART bus interface: Up to 2

Figure 5-9 Gowin DC-SCM LTPI IP Configuration Interface 8

The screenshot shows the 'OEM' configuration tab. It contains two settings: 'OEM Data Width(1-32)' set to 32 and 'OEM Capability 0 in HEX(0x)' set to 0000.

Setting	Value
OEM Data Width(1-32)	32
OEM Capability 0 in HEX(0x)	0000

- OEM Data Width: OEM data, up to 32bit.
- OEM Capability: Fill fields only, no other meaning at present.

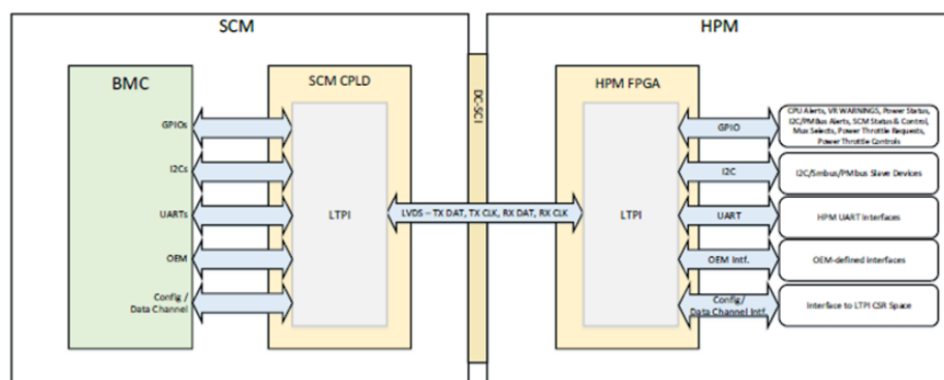
6 Reference Design

See Gowin DC-SCM LTPI IP reference design for details at Gowinsemi website.

6.1 Application

The DC-SCM LTPI is designed primarily to transmit low-speed signals between HPM and SCM.

Figure 6-1 HPM and SCM Connection

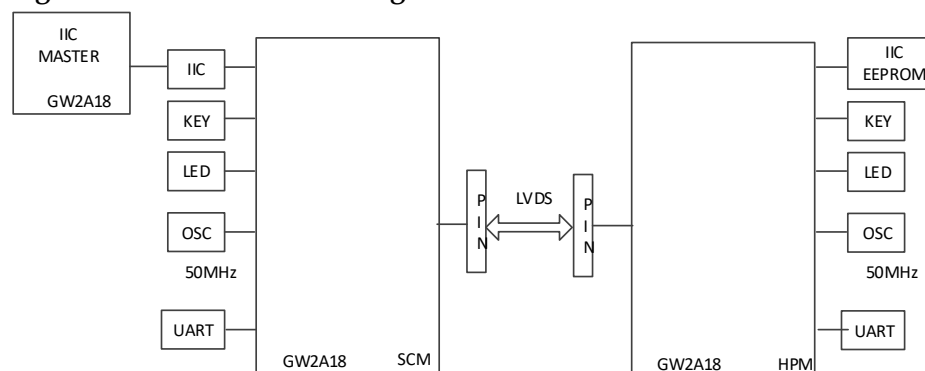


6.2 Reference Design

This chapter is intended to introduce the example and using of Gowin DC-SCM LTPI IP reference design.

6.2.1 Hardware Platform

Figure 6-2 Platform Block Diagram

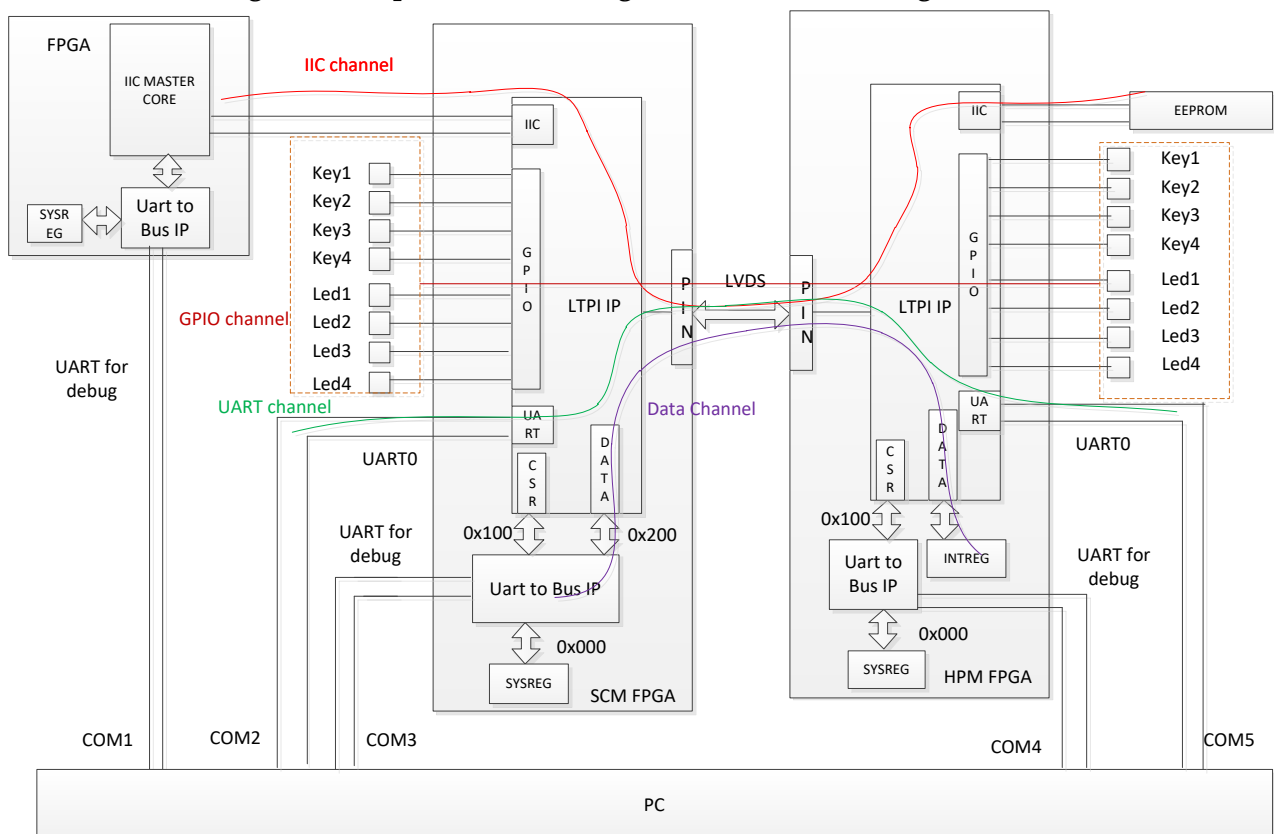


One such reference example is implemented on 3 GW2A18 boards with the following hardware conditions:

1. There is an on-board 50MHz crystal oscillator for generating system clocks and reference clocks, and bus configuration clocks, etc.
2. On-board KEY and LED are used for testing LLGPIO and NLGPIO channels.
3. Pin headers are used to connect UART to a PC for testing UART channels.
4. The FPGA board as SCM is externally connected to another GW2A18 board for simulating an IIC MASTER. Meanwhile, the FPGA board as HPM is externally connected to EEPROM with IIC interfaces, acting as an IIC SLAVE.
5. Test the Data channel by simulating APB timing through UART interface and UART to Bus IP.

6.2.2 Implementation Diagram

Figure 6-3 Implementation Diagram of Reference Design



The main modules are as follows:

- DC-SCM LTPI IP Module

The configuration of this module is shown from Figure 5-2 to Figure 5-9.

- LLGPIO is set to 16

- NLGPIO is set to 32
OEM Data width is set to 32.
- The number of IICs/SMBus is set to 1
- The number of UART is set to 1

The key and LED mapping routes for testing GPIO and OEM channels are shown in the following table:

Table 6-1 Key and LED Mapping Route

Start point	Route	End point
KEY1	LLGPIO[0]	LED1
KEY2	NLGPIO[0]	LED2
KEY3	NLGPIO[31]	LED3
KEY4	OEM[0]	LED3

Tests show that GPIO and OEM channels are bi-directional. LED1 on HPM FPGA lights up when KEY1 on SCM FPGA is pressed; LED1 on SCM FPGA lights up when KEY1 on HPM FPGA is pressed.

- UART_to_Bus module: Serial to bus module, converting serial commands sent by serial assistant or master into bus control timing for accessing CSR registers and testing data channels
- sysreg module: Generate local system configuration registers

6.2.3 Bus Protocol & Address

The custom serial bus protocol conventions of UART_to_Bus IP module are as follows:

Write bus **"W AAAA BBBBBBBB"**

Read bus **"R AAAA"**

The read bus will return the following command **"G AAAA BBBBBBBB"**

See [IPUG1022, Gowin UART to Bus IP User Guide](#) for the specific use of UART to Bus IP.

The detailed register address definitions in the reference project are as follows:

Table 6-2 Register Definition

Base Address	Description
0x000	System registers used to control FPGA project
	0x00[0] 0: Enable LTPI, 1: Disable LTPI
	0x01[7:0] Data channel Tag field input
	0x81[0] Configurable PLL lock indicator 1: Locked 0: Unlocked
	0x82[0] Byte alignment indicator

Base Address	Description	
		1: Aligned 0: Misaligned
	0x84[7:0]	Data channel Tag field output
0x100	Used to connect to the LTPI CSR interface	
0x200	SCM: Used to connect to the LTPI Data channel interface HPM: There is no such base address	

6.3 On-Board Testing

6.3.1 Operation Description

The sample project is tested on the DK_START_GW2A18 V2.0 development board. If you don't have this board, but you want to run the reference project, you just need to modify the serial port pins and clock input pins according to the actual situation of your board for testing.

After the modification, the bit file is generated and download it to the FPGA to start testing.

6.3.2 Operating Flow

1. Connect Cables

- Connect the LVDS cables between SCM and HPM, note that the interface shall meet the electrical specifications for the LVDS interface.
- The FPGA as IIC Master is connected to the SCM FPGA, and the SDA and SCL of IIC need to be pulled up on the hardware, note that the interface shall meet the electrical specifications of the IIC interface.
- Connect the serial interface to the PC and record the serial number of each serial port for easy debugging.

2. Power on 3 FPGA Boards and Query LTPI State

By default, SCM and HPM will start operating when powered on, and at this point, you can query the internal state of LTPI by connecting to the serial port of the CSR registers.

Query the CSR register of SCM through COM3 and the CSR register of HPM through COM4 (the specific COM numbers need to be queried through PC Device Manager) with the baud rate of 460800bps, the data of 8 bits, the stop bit of 1bit, and no parity bit.

Send instruction R 81, feedback G 0081 00000001, indicating internal PLL lock

Send instruction R 82, feedback G 0082 00000001, indicating byte alignment

Send instruction R 100, feedback G 0100 00044481

According to the CSR register definition, both SCM and HPM are already in Operation Mode.

3. Test GPIO and OEM Channels

When you press KEY1~KEY4 on SCM or HPM, the corresponding LED1~LED4 will light on, indicating that the GPIO and OEM channels are normal.

4. Test UART Channels

Connect COM2 and COM5 serial ports corresponding to the UART channels of SCM and HPM, (the specific COM numbers need to be queried through PC Device Manager), with both sides configured at a baud rate of 115200. If anything is sent on both sides and received correctly on both side, it indicates that UART channel is normal.

5. Test Data Channels

The Data channel converts serial instructions into APB bus timing via UART to Bus IP. The mapping base address is 0x200.

Enter W 200 12345678 in the debugging serial port COM3 of the SCM, and then read back R 200. If the result returned back is G 0200 12345678, it indicates that the Data channel is normal.

6. Test IIC Channel

Testing IIC channel requires an IIC master controller outside the SCM to issue IIC timing and an IIC slave device to be connected to the HPM. The EEPROM device is selected here. So there is also a board dedicated to the IIC master controller, which is implemented with the Gowin IIC Master IP. See [IPUG504, Gowin IIC Master IP User Guide](#) for the specific use. UART to Bus IP is used for debugging here, and the 0x100 base address is connected to the internal bus port of the IIC Master IP.

The serial port used here is COM1 (the specific COM number needs to be queried through the PC Device Manager), with the baud rate of 460800bps, the data of 8 bits, the stop bit of 1bit, and no parity bit.

The debugging instructions are as follows:

1. First, write 0x11 to the 0x01 address of the IIC EEPROM, and 0x12 to 0x02, the instructions are as follows:

W 0100 c8 — Set IIC rate

W 0101 00 — Set IIC rate

R 0100 — Read back IIC rate and return the value G 0100 000000C8

W 0102 80 — Start IIC core

W 0103 A0 — Set IIC device address and set to Write

W 0104 90 — Generate start timing and write device address timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W0103 01 — Set IIC device internal address 0x01

W 0104 10 — Write internal address 0x01 and generate write internal address timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W 0103 11 — Write 0x11 to the current IIC address (0x01)

W 0104 10 — Generate write data timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W 0103 12 — Write 0x12 to next IIC address (0x02, auto-increment)

W 0104 50 — Generate write and stop timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000001

2. Read back the 0x01, 0x02 addresses of the IIC EEPROM, the instructions are as follows:

W 0100 c8 — Set IIC rate

W 0101 00 — Set IIC rate

R 0100 — Read back IIC rate and return the value G 0100 000000C8

W 0102 80 — Start IIC core

W 0103 A0 — Set IIC device address and set to Write

W 0104 90 — Generate start timing and write device address timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W 0103 01 — Set IIC device internal address 0x01

W 0104 10 — Write internal address 0x01 and generate write internal address timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W 0103 A1 — Set IIC device address and set to read operation

W 0104 90 — Generate start timing and write address timing

R 0104 — Wait for the operation to complete and return the value G 0104 00000041

W 0104 20 — Read current IIC address (0x01) data

R 0103 — Wait for the return value to return the instruction G 0103 00000011

W 0104 28 — Last read IIC address (0x02) data

R 0103 — Wait for the return value to return the instruction G

0103 00000012

W 0104 40 ——— Generate stop timing

R 0104 ——— Wait for the operation to complete and return the
instruction G 0104 00000081

7 File Delivery

The delivery files for the Gowin DC-SCM LTPI IP includes the documents, the design source code, and the reference design.

7.1 Documentation

The folder mainly contains the user guide in PDF version.

Table 7-1 Documents List

Name	Description
IPUG1086, Gowin DC-SCM LTPI IP User Guide	Gowin DC-SCM LTPI IP User Guide, namely this one

7.2 Design Source Code (Encryption)

The Encryption Code Folder contains the RTL encryption code of Gowin DC-SCM LTPI IP used for the GUI, to generate the IP core as needed.

Table 7-2 Gowin DC-SCM LTPI IP Design Source Code List

Name	Description
ltpi_core.v	The top-level file of the IP core, which provides users with interface information, encrypted.

7.3 Reference Design

Gowin DC-SCM LTPI IP RefDesign folder contains the netlist file, user reference design, constraints file, top-level file, and project file, etc.

Table 7-3 Gowin DC-SCM LTPI IP RefDesign Folder Content List

Name	Description
top.v	The top module of reference design
top.cst	Project physical constraints file
top.sdc	Project timing constraints file
dcscm_ltpi.v	Generate DC-SCM LTPI IP top-level file, encrypted.
uart_to_bus.v	Serial to bus module, encrypted
sysreg.v	SysReg generation module
syspll.v	PLL IP file

