



Arora V Hardened MIPI D-PHY

User Guide

UG296-1.0E, 9/28/2023

Copyright © 2023 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN, Gowin and GOWINSEMI are trademarks of Guangdong Gowin

Semiconductor Corporation and are 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
09/28/2023	1.0E	Initial version published.

Contents

Contents	iv
List of Figures	vi
List of Tables	vii
1 About This Guide	1
1.1 Purpose	1
1.2 Related Documents	1
1.3 Terminology and Abbreviations.....	2
1.4 Support and Feedback	2
2 Overview	3
2.1 Features.....	3
2.2 Functional Description	4
2.2.1 MIPI D-PHY RX	4
2.2.2 MIPI D-PHY TX.....	5
3 MIPI D-PHY Primitives	6
3.1 MIPI D-PHY	6
3.1.1 Port Description	6
3.1.2 Parameter Description	13
3.1.3 Primitive Instantiation	14
3.2 MIPI D-PHY RX	24
3.2.1 Port Description	24
4 IP Generation	27
4.1 MIPI D-PHY TX IP	27
4.1.1 Port Description	27
4.1.2 Typical Timing	29
4.1.3 Configuration Description	30

4.2 MIPI D-PHY RX IP	32
4.2.1 Port Description	32
4.2.2 Typical Timing	33
4.2.3 Configuration Description	34
Appendix A MIPI D-PHY Rate Table	36

List of Figures

Figure 2-1 MIPI D- PHY RX Structure View.....	4
Figure 2-2 MIPI D- PHY TX Structure View	5
Figure 4-1 MIPI D-PHY TX Typical Timing.....	29
Figure 4-2 MIPI D- PHY Configuration (TX).....	30
Figure 4-3 MIPI D-PHY RX Typical Timing	33
Figure 4-4 MIPI D- PHY Configuration (RX).....	34

List of Tables

Table 1-1 Terminology and Abbreviations	2
Table 3-1 MIPI D-PHY Port Description	6
Table 3-2 MIPI D-PHY Parameters	13
Table 3-3 MIPI D-PHY RX Ports	24
Table 4-1 MIPI D-PHY TX IP Ports	27
Table 4-2 MIPI D-PHY RX IP Ports	32
Table A-1 MIPI D-PHY Rates (Arora Family)	37

1 About This Guide

1.1 Purpose

Arora V MIPI D-PHY guide is mainly used help you quickly learn the features and usage of Arora V MIPI- PHY by introducing to the functions, ports, configuration, etc.

1.2 Related Documents

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

- [DS981E, GW5AT series of FPGA Products Data Sheet](#)
- [DS1103E, GW5A series of FPGA Products Data Sheet](#)
- [DS981E, GW5AST series of FPGA Products Data Sheet](#)
- [DS1114E, GW5AS-138 Data Sheet](#)
- [DS1108E, GW5AR series of FPGA Products Data Sheet](#)
- [DS1113E, GW5A series of FPGA Products \(Automotive operation\) Data Sheet](#)
- [DS1111E, GW5AT series of FPGA Products \(Automotive operation\) Data Sheet](#)

1.3 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Full Name
CSI	Camera Serial Interface
DSI	Display Serial Interface
HS	High Speed
I/O	Input/output
LP	Low Power
MIPI	Mobile Industry Processor Interface

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 Overview

Arora V devices contain the hardcore MIPI D-PHY RX and MIPI D-PHY PHY^[1], and is compliant with the MIPI Alliance Standard for D-PHY Specification V2.1. The dedicated D-PHY core supports MIPI DSI and CSI-2 mobile video interfaces for cameras and displays.

Note!

[1] GOWIN 138K devices support MIPI D-PHY RX. GOWIN 25K devices support MIPI D-PHY RX and MIPI D-PHY TX.

2.1 Features

The main features of MIPI D- PHY are as follows:

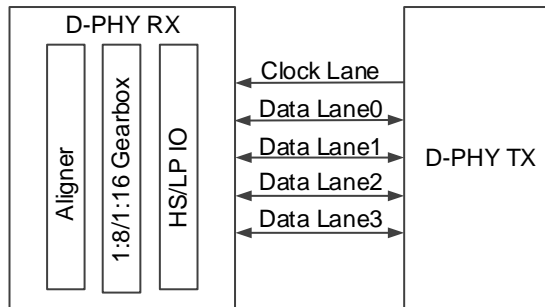
- Unidirectional HS (High-speed) mode at up to 2.5 Gbps per lane and 10 Gbps per quad (4 data lanes). Up to 20 Gbps supported by each chip (8 data lanes in all).
- Supports two MIPI D-PHY quads, up to 4 data lanes and one clock lane for each quad.
- Bidirectional Low-power (LP) mode with a bit rate of 10Mbps.
- Built-in HS Sync, bit alignment in the lane (Word Alignment) and word alignment between lanes (Lane Alignment).
- MIPI D-PHY RX supports 1:8 mode and 1:16 mode.
- MIPI D- PHY TX supports 8:1 mode and 16:1 mode.
- Supports MIPI DSI and MIPI CSI-2 link layers.
- A dedicated MIPI Bank for the hardcore MIPI D-PHY.
- GW5AT-138 supports two MIPI D-PHY RX quads.
- GW5A-25 supports one MIPI D-PHY quad, including RX and TX.

2.2 Functional Description

2.2.1 MIPI D-PHY RX

MIPI D-PHY RX includes three parts: HS/ IO, 1:8/1:16 Gearbox and Aligner, as shown in Figure 2-1.

Figure 2-1 MIPI D- PHY RX Structure View



HS/LP I/O

Supports dynamic ODT, and dynamic LP TX/RX and HS RX.

1:8/1:16 Gearbox

8-bit or 16-bit width is supported and the primitive instantiation can be configured via the parameter HS_8BIT_MODE.

Aligner

Word align and Lane align are supported and can be configured via WALIGN_BY and LALIGN_EN ports. The key of Word align can be user-defined (ALIGN_BYTE), supporting 2 bytes and 3 bytes mode, which is configured via IPI_DPHY_RX port or ONE_BYTE0_MATCH port.

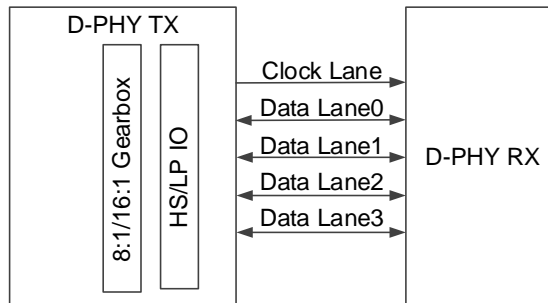
Note!

For the MIPI_DPHY_RX port description, please refer to 3 MIPI D-PHY Primitives > 3.1.1 Port Description, or 4.2 MIPI D-PHY RX IP > 4.2.1 Port Description.

2.2.2 MIPI D-PHY TX

MIPI D-PHY TX includes two parts: HS/LP IO and 1:16/ 16: 1 Gearbox, as shown in Figure 2-1.

Figure 2-2 MIPI D- PHY TX Structure View



HS/LP I/O

Supports dynamic LP TX/RX and HS RX.

1:16/ 16: 1 Gearbox

8-bit or 16-bit width is supported and can be configured via HS_8BIT_MODE.

Note !

For the MIPI_DPHY_RX port description, please refer to 3. MIPI D-PHY Primitives > 3.1.1 Port Description.

3 MIPI D-PHY Primitives

3.1 MIPI D-PHY

3.1.1 Port Description

Table 3-1 MIPI D-PHY Port Description

Port	I/O	Description
MIPI INTERFACE Signals		
CK_N	inout	CK Lane Complement Input
CK_P	inout	CK Lane True Input
D<0~3>_N	inout	Data Lane <0~3> Complement Input
D<0~3>_P	inout	Data Lane <0~3> True Input
RESET Signals		
RESET	input	Reset signal: 1'b1: reset all;
RX_DRST_N	input	RX digital reset, active low
TX_DRST_N	input	TX digital reset, active low
PWRON_RX	input	HSRX Power On Control: <ul style="list-style-type: none"> ● 1'b1: HSRX on ● 1'b0: HSRX off to standby in low power state
PWRON_TX	input	HSTX Power On Control: <ul style="list-style-type: none"> ● 1'b1: HSTX on ● 1'b0: HSTX off to standby in low power state
HSRX_STOP	input	HSRX Clock Stop Signal for synchronization
WALIGN_DVLD	input	word aligner input data valid from the fabric
CLOCK Signals		
CK0	input	HSTX: ck0

Port	I/O	Description
CK90	input	HSTX: ck90
CK180	input	HSTX: ck180
CK270	input	HSTX: ck270
RX_CLK_O	output	HSRX output 1X clock, max 93.75MHz@1.5Gbps
TX_CLK_O	output	HSTX output 1X clock, max 93.75MHz@1.5Gbps
RX_CLK_1X	input	1X clock from fabric, max 93.75MHz@1.5Gbps
TX_CLK_1X	input	1X clock from fabric, max 93.75MHz@1.5Gbps
HSRX Signals		
D0LN_HSRXD	output	data lane0 HS data output to fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D1LN_HSRXD	output	data lane1 HS data output to fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D2LN_HSRXD	output	data lane2 HS data output to fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D3LN_HSRXD	output	data lane3 HS data output to fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D0LN_HSRXD_VLD	output	data lane0 HS data output valid to fabric
D1LN_HSRXD_VLD	output	data lane1 HS data output valid to fabric
D2LN_HSRXD_VLD	output	data lane2 HS data output valid to fabric
D3LN_HSRXD_VLD	output	data lane3 HS data output valid to fabric
HSRX_EN_CK	input	CK Lane: 1'b1 HSRX enabled
HSRX_EN_D0	input	Data Lane0: ● 1'b1: HSRX enabled ● 1'b0: HSRX disabled
HSRX_EN_D1	input	Data Lane1: ● 1'b1: HSRX enabled ● 1'b0: HSRX disabled
HSRX_EN_D2	input	Data Lane2: ● 1'b1: HSRX enabled ● 1'b0: HSRX disabled
HSRX_EN_D3	input	Data Lane3:

Port	I/O	Description
		<ul style="list-style-type: none"> ● 1'b1: HSRX enabled ● 1'b0: HSRX disabled
HSRX_ODTEN_CK	input	CK HSRX ODT enabled: <ul style="list-style-type: none"> ● 1'b1 ODT enabled ● 1'b0 ODT disabled
HSRX_ODTEN_D0	input	Data Lane0 HSRX ODT enabled: <ul style="list-style-type: none"> ● 1'b1 ODT enabled ● 1'b0 ODT disabled
HSRX_ODTEN_D1	input	Data Lane1 HSRX ODT enabled: <ul style="list-style-type: none"> ● 1'b1 ODT enabled ● 1'b0 ODT disabled
HSRX_ODTEN_D2	input	Data Lane2 HSRX ODT enabled: <ul style="list-style-type: none"> ● 1'b1 ODT enabled ● 1'b0 ODT disabled
HSRX_ODTEN_D3	input	Data Lane3 HSRX ODT enabled: <ul style="list-style-type: none"> ● 1'b1 ODT enabled ● 1'b0 ODT disabled
D0LN_HSRX_DREN	input	Data Lane0 HSRX driver enabled: <ul style="list-style-type: none"> ● 1'b1 driver enabled ● 1'b0 driver disabled
D1LN_HSRX_DREN	input	Data Lane1 HSRX driver enabled: <ul style="list-style-type: none"> ● 1'b1 driver enabled ● 1'b0 driver disabled
D2LN_HSRX_DREN	input	Data Lane2 HSRX driver enabled: <ul style="list-style-type: none"> ● 1'b1 driver enabled ● 1'b0 driver disabled
D3LN_HSRX_DREN	input	Data Lane3 HSRX driver enabled <ul style="list-style-type: none"> ● 1'b1 driver enabled ● 1'b0 driver disabled
HSRX_DLYDIR_LANE0	input	Data Lane0: Direction for HSRX Deskew Delay Control: 0 Count Up; 1 Count Down;
HSRX_DLYDIR_LANE1	input	Data Lane1: Direction for HSRX Deskew Delay Contr Data Lane0: Direction for HSRX Deskew
HSRX_DLYDIR_LANE2	input	Data Lane2: Direction for HSRX Deskew Delay Control: 0 Count Up; 1 Count Down;
HSRX_DLYDIR_LANE3	input	Data Lane3: Direction for HSRX Deskew Delay Control: 0 Count Up; 1 Count Down;

Port	I/O	Description
HSRX_DLYDIR_LANECK	input	CK Lane: Direction for HSRX Deskew Delay Control: 0 Count Up; 1 Count Down
HSRX_DLYLDN_LANE0	input	Data Lane0: Load HSRX Deskew Delay Control input from Fuse: 1'b0 load
HSRX_DLYLDN_LANE1	input	Data Lane1: Load HSRX Deskew Delay Control input from Fuse: 1'b0 load
HSRX_DLYLDN_LANE2	input	Data Lane2: Load HSRX Deskew Delay Control input from Fuse: 1'b0 load
HSRX_DLYLDN_LANE3	input	Data Lane3: Load HSRX Deskew Delay Control input from Fuse: 1'b0 load
HSRX_DLYLDN_LANECK	input	CK Lane: Load HSRX Deskew Delay Control input from Fuse: 1'b0 load;
HSRX_DLYMV_LANE0	input	Data Lane0: enable HSRX Deskew Delay Control to count: 1'b1 move
HSRX_DLYMV_LANE1	input	Data Lane1: enable HSRX Deskew Delay Control to count: 1'b1 move
HSRX_DLYMV_LANE2	input	Data Lane2: enable HSRX Deskew Delay Control to count: 1'b1 move
HSRX_DLYMV_LANE3	input	Data Lane3: enable HSRX Deskew Delay Control to count: 1'b1 move
HSRX_DLYMV_LANECK	input	CK Lane: enable HSRX Deskew Delay Control to count: 1'b1 move
D0LN_DESKEW_DONE	output	D0ln_deskew_done
D1LN_DESKEW_DONE	output	D1ln_deskew_done
D2LN_DESKEW_DONE	output	D2ln_deskew_done
D3LN_DESKEW_DONE	output	D3ln_deskew_done
D0LN_DESKEW_ERROR	output	D0ln_deskew_error
D1LN_DESKEW_ERROR	output	D1ln_deskew_error
D2LN_DESKEW_ERROR	output	D2ln_deskew_error
D3LN_DESKEW_ERROR	output	D3ln_deskew_error
D0LN_DESKEW_REQ	input	D0lane deskew function request
D1LN_DESKEW_REQ	input	D1lane deskew function request
D2LN_DESKEW_REQ	input	D2lane deskew function request
D3LN_DESKEW_REQ	input	D3lane deskew function request
HSTX Signals		
CKLN_HSTXD	input	CK lane0 HS data input from fabric

Port	I/O	Description
		1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D0LN_HSTXD	input	data lane0 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D1LN_HSTXD	input	data lane1 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D2LN_HSTXD	input	data lane2 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D3LN_HSTXD	input	data lane3 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
HSTXD_VLD	input	HS_TX Data Valid input from fabric
HSTXEN_LNCK	input	CK Lane0: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN0	input	Data Lane0: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN1	input	Data Lane1: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN2	input	Data Lane2: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN3	input	Data Lane3: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
TXDPEN_LN0	input	txdpen_in0,1'b1 enabled
TXDPEN_LN1	input	txdpen_in1,1'b1 enabled
TXDPEN_LN2	input	txdpen_in2,1'b1 enabled
TXDPEN_LN3	input	txdpen_in3,1'b1 enabled
TXDPEN_LNCK	input	txdpen_inck,1'b1 enabled
TXHCLK_EN	input	txhclk_en,1'b1 enabled

Port	I/O	Description
LPRX Signals		
DI_LPRX0_N	output	Data Lane0 Complement Pad LPRX input
DI_LPRX0_P	output	Data Lane0 True Pad LPRX input
DI_LPRX1_N	output	Data Lane1 Complement Pad LPRX input
DI_LPRX1_P	output	Data Lane1 True Pad LPRX input
DI_LPRX2_N	output	Data Lane2 Complement Pad LPRX input
DI_LPRX2_P	output	Data Lane2 True Pad LPRX input
DI_LPRX3_N	output	Data Lane3 Complement Pad LPRX input
DI_LPRX3_P	output	Data Lane3 True Pad LPRX input
DI_LPRXCK_N	output	CK Lane Complement Pad LPRX input
DI_LPRXCK_P	output	CK Lane True Pad LPRX input
LPRX_EN_CK	input	CK Lane: 1'b1 LPRX enabled
LPRX_EN_D0	input	Data Lane0: <ul style="list-style-type: none"> ● 1'b1 LPRX enabled ● 1'b0 LPRX disabled
LPRX_EN_D1	input	Data Lane1: <ul style="list-style-type: none"> ● 1'b1 LPRX enabled ● 1'b0 LPRX disabled
LPRX_EN_D2	input	Data Lane2: <ul style="list-style-type: none"> ● 1'b1 LPRX enabled ● 1'b0 LPRX disabled
LPRX_EN_D3	input	Data Lane3: <ul style="list-style-type: none"> ● 1'b1 LPRX enabled ● 1'b0 LPRX disabled
LPTX Signals		
DO_LPTX0_N	input	Data Lane0 Complement Pad LPTX output
DO_LPTX0_P	input	Data Lane0 True Pad LPTX output
DO_LPTX1_N	input	Data Lane1 Complement Pad LPTX output
DO_LPTX1_P	input	Data Lane1 True Pad LPTX output
DO_LPTX2_N	input	Data Lane2 Complement Pad LPTX output
DO_LPTX2_P	input	Data Lane2 True Pad LPTX output
DO_LPTX3_N	input	Data Lane3 Complement Pad LPTX output
DO_LPTX3_P	input	Data Lane3 True Pad LPTX output
Do_LPTXCK_n	input	CK Lane Completion Pad LPTX output

Port	I/O	Description
Do_LPTXCK_P.	input	CK Lane True Pad LPTX output
LPTXEN_LNCK	input	CK Lane: 1'b1 LPTX enabled
LPTXEN_LN0	input	Data Lane0: 1'b1 LPTX enabled
LPTXEN_LN1	input	Data Lane1: 1'b1 LPTX enabled
LPTXEN_LN2	input	Data Lane2: 1'b1 LPTX enabled
LPTXEN_LN3	input	Data Lane3: 1'b1 LPTX enabled
ALP Signals		
ALPEDO_LANE0	output	ALP Mode: Data Lane0 output
ALPEDO_LANE1	output	ALP Mode: Data Lane1 output
ALPEDO_LANE2	output	ALP Mode: Data Lane2 output
ALPEDO_LANE3	output	ALP Mode: Data Lane3 output
ALPEDO_LANECK	output	ALP Mode: CK Lane output
ALP_EDEN_LANE0	input	ALP Mode: alp_eden_lane0
ALP_EDEN_LANE1	input	ALP Mode: alp_eden_lane1
ALP_EDEN_LANE2	input	ALP Mode: alp_eden_lane2
ALP_EDEN_LANE3	input	ALP Mode: alp_eden_lane3
Alp_eden_LANECK	input	ALP Mode: Alp_eden_laneck
ALPEN_LN0	input	ALP Mode: 1'b1, Lane0 enabled
ALPEN_LN1	input	ALP Mode: 1'b1, Lane1 enabled
ALPEN_LN2	input	ALP Mode: 1'b1, Lane2 enabled
ALPEN_LN3	input	ALP Mode: 1'b1, Lane3 enabled
ALPEN_LNCK	input	ALP Mode: 1'b1, CK lane enabled
MRDATA [7:0]	output	Mrdata
MA_INC	input	ma_inc
MCLK	input	mclk
MOPCODE	input	Mopcode
MWDATA [7:0]	input	Mwdata

3.1.2 Parameter Description

Table 3-2 MIPI D-PHY Parameters

Parameter	Default Value	Description
RX_ALIGN_BYTE	8'b10111000	KEY for word aligner and lane aligner
RX_byte_LITTLE_ENDIAN	1'b1	1'b1: Littlendian
RX_CLK_1X_SYNC_SEL	1'b0	Select clock source for HS lane output data: <ul style="list-style-type: none"> ● 0: Select fabric input clock rx_clk_1x ● 1: Select output clock rx_clk_o
RX_HS_8BIT_MODE	1'b0	1'b1:8bit mode; 1'b0:16bit mode
RX_INVERT	1'b0	Data density selection: 1'b1 invert
RX_LANE_ALIGN_en	1'b0	1'b1:lane aligner enable
RX_ONE_BYTE0_MATCH	1'b0	Byte count match in word aligner
RX_RD_START_DEPTH	5'b00001	
RX_SYNC_MODE	1'b0	
RX_world_alarm_bypass	1'b0	
RX_WORD_ALIGN_data_VLD_SRC_SEL	1'b0	
RX_world_LITTLE_ENDIAN	1'b1	1'b1: Little endian of dual word (8bit/word). Not used in 8bit data output mode
TX_BYPASS_MODE	1'b0	
TX_BYTECLK_SYNC_MODE	1'b0	
TX_HS_8BIT_MODE	1'b0	1'b1:8bit mode; 1'b0:16bit mode
TX_OCLK_USE_CIBCLK	1'b0	
TX_RD_START_DEPTH	5'b00001	
TX_SYNC_MODE	1'b0	
TX_WORD_LITTLE_ENDIAN	1'b1	1'b1: Little endian of dual word (8bit/word). Not used in 8bit data output mode

3.1.3 Primitive Instantiation

Verilog Instantiation:

```

MIPI_DPHY mipi_dhpy_inst (
    .ALPEDO_LANE0(alpedo_lane0),
    .ALPEDO_LANE1(alpedo_lane1),
    .ALPEDO_LANE2(alpedo_lane2),
    .ALPEDO_LANE3(alpedo_lane3),
    .ALPEDO_LANECK(alpedo_laneck),
    .RX_CLK_O(rx_clk_o)
    .TX_CLK_O(tx_clk_o),
    .D0LN_DESKEW_DONE(d0ln_deskew_done),
    .D1LN_DESKEW_DONE(d1ln_deskew_done),
    .D2LN_DESKEW_DONE(d2ln_deskew_done),
    .D3LN_DESKEW_DONE(d3ln_deskew_done),
    .D0LN_DESKEW_ERROR(d0ln_deskew_error),
    .D1LN_DESKEW_ERROR(d1ln_deskew_error),
    .D2LN_DESKEW_ERROR(d2ln_deskew_error),
    .D3LN_DESKEW_ERROR(d3ln_deskew_error),
    .D0LN_HSRXD(d0ln_hsrxd),
    .D1LN_HSRXD(d1ln_hsrxd),
    .D2LN_HSRXD(d2ln_hsrxd),
    .D3LN_HSRXD(d3ln_hsrxd),
    .D0LN_HSRXD_VLD(d0ln_hsrxd_vld),
    .D1LN_HSRXD_VLD(d1ln_hsrxd_vld),
    .D2LN_HSRXD_VLD(d2ln_hsrxd_vld),
    .D3LN_HSRXD_VLD(d3ln_hsrxd_vld),
    .DI_LPRX0_N(di_lprx0_n),
    .DI_LPRX0_P(di_lprx0_p),
    .DI_LPRX1_N(di_lprx1_n),
    .DI_LPRX1_P(di_lprx1_p),
    .DI_LPRX2_N(di_lprx2_n),
    .DI_LPRX2_P(di_lprx2_p),
    .DI_LPRX3_N(di_lprx3_n),
    .DI_LPRX3_P(di_lprx3_p),
    .DI_LPRXCK_N(di_lprxck_n),
    .DI_LPRXCK_P(di_lprxck_p),
    MRDATA (mrdata),
    .CK_N (ck_n),
    .CK_P (ck_p),
    .D0_N(d0_n),
    .D0_P(d0_p),
    .D1_N(d1_n),
    .D1_P(d1_p),

```

.D2_N(d2_n),
.D2_P(d2_p),
.D3_N(d3_n),
.D3_P(d3_p),
.ALP_EDEN_LANE0(alp_eden_lane0),
.ALP_EDEN_LANE1(alp_eden_lane1),
.ALP_EDEN_LANE2(alp_eden_lane2),
.ALP_EDEN_LANE3(alp_eden_lane3),
.ALP_EDEN_LANECK(alp_eden_laneck),
.ALPEN_LN0(alpen_ln0),
.ALPEN_LN1(alpen_ln1),
.ALPEN_LN2(alpen_ln2),
.ALPEN_LN3(alpen_ln3),
.ALPEN_LNCK(alpen_lnck),
.HSRX_STOP (hsrx_stop),
.HSTXEN_LN0(hstxen_ln0),
.HSTXEN_LN1(hstxen_ln1),
.HSTXEN_LN2(hstxen_ln2),
.HSTXEN_LN3(hstxen_ln3),
.HSTXEN_LNCK (hstxen_lnck),
.LPTXEN_LN0(lptxen_ln0),
.LPTXEN_LN1(lptxen_ln1),
.LPTXEN_LN2(lptxen_ln2),
.LPTXEN_LN3(lptxen_ln3),
.LPTXEN_LNCK (lptxen_lnck),
.PWRON_RX (pwron_rx),
.PWRON_TX (pwron_tx),
.RESET(reset),
.RX_CLK_1X(rx_clk_1x),
.TX_CLK_1X(tx_clk_1x),
.TXDPEN_LN0(txdpenn_in0),
.TXDPEN_LN1(txdpenn_in1),
.TXDPEN_LN2(txdpenn_in2),
.TXDPEN_LN3(txdpenn_in3),
.TXDPEN_LNCK(txdpenn_inck),
.TXHCLK_EN(txhclk_en),
.CKLN_HSTXD (ckln_hstxd),
.D0LN_HSTXD(d0ln_hstxd),
.D1LN_HSTXD(d1ln_hstxd),
.D2LN_HSTXD(d2ln_hstxd),
.D3LN_HSTXD(d3ln_hstxd),
.HSTXD_VLD (hstxd_vld),
.CK0(ck0),
.CK90(ck90),

.CK180(ck180),
.CK270(ck270),
.D0LN_DESKEW_REQ(d0ln_deskew_req),
.D1LN_DESKEW_REQ(d1ln_deskew_req),
.D2LN_DESKEW_REQ(d2ln_deskew_req),
.D3LN_DESKEW_REQ(d3ln_deskew_req),
.D0LN_HSRX_DREN(d0ln_hsrx_dren),
.D1LN_HSRX_DREN(d1ln_hsrx_dren),
.D2LN_HSRX_DREN(d2ln_hsrx_dren),
.D3LN_HSRX_DREN(d3ln_hsrx_dren),
.DO_LPTX0_N(do_lptx0_n),
.DO_LPTX0_P(do_lptx0_p),
.DO_LPTX1_N(do_lptx1_n),
.DO_LPTX1_P(do_lptx1_p),
.DO_LPTX2_N(do_lptx2_n),
.DO_LPTX2_P(do_lptx2_p),
.DO_LPTX3_N(do_lptx3_n),
.DO_LPTX3_P(do_lptx3_p),
.DO_LPTXCK_N(do_lptxck_n),
.DO_LPTXCK_P(do_lptxck_p),
.HSRX_DLYDIR_LANE0(hsrx_dlydir_lane0),
.HSRX_DLYDIR_LANE1(hsrx_dlydir_lane1),
.HSRX_DLYDIR_LANE2(hsrx_dlydir_lane2),
.HSRX_DLYDIR_LANE3(hsrx_dlydir_lane3),
.HSRX_DLYDIR_LANECK (hsrx_dlydir_laneck)
.HSRX_DLYLDN_LANE0(hsrx_dlyldn_lane0),
.HSRX_DLYLDN_LANE1(hsrx_dlyldn_lane1),
.HSRX_DLYLDN_LANE2(hsrx_dlyldn_lane2),
.HSRX_DLYLDN_LANE3(hsrx_dlyldn_lane3),
.HSRX_DLYLDN_LANECK (hsrx_dlyldn_laneck)
.HSRX_DLYMV_LANE0(hsrx_dlymv_lane0),
.HSRX_DLYMV_LANE1(hsrx_dlymv_lane1),
.HSRX_DLYMV_LANE2(hsrx_dlymv_lane2),
.HSRX_DLYMV_LANE3(hsrx_dlymv_lane3),
.HSRX_DLYMV_LANECK (hsrx_dlymv_laneck)
.HSRX_EN_CK (hsrx_en_ck)
.HSRX_EN_D0(hsrx_en_d0),
.HSRX_EN_D1(hsrx_en_d1),
.HSRX_EN_D2(hsrx_en_d2),
.HSRX_EN_D3(hsrx_en_d3),
.HSRX_ODTEN_CK (hsrx_odten_ck),
.HSRX_ODTEN_D0(hsrx_odten_d0),
.HSRX_ODTEN_D1(hsrx_odten_d1),
.HSRX_ODTEN_D2(hsrx_odten_d2),

```

.HSRX_ODTEN_D3(hsrx_odten_d3),
.LPRX_EN_CK (lprx_en_ck),
.LPRX_EN_D0(lprx_en_d0),
.LPRX_EN_D1(lprx_en_d1),
.LPRX_EN_D2(lprx_en_d2),
.LPRX_EN_D3(lprx_en_d3),
.MA_INC(ma_inc),
.MCLK(mclk),
.MOPCODE(mopcode),
.MWDATA(mwdata),
.RX_DRST_N(rx_drst_n)
.TX_DRST_N(tx_drst_n),
.WALIGN_DVLD(walign_dvld)
);

```

```

defparam mipi_dhpy_inst.TX_PLLCLK = "NONE";
defparam mipi_dhpy_inst.CKLN_DELAY_EN = 1'b0;
defparam mipi_dhpy_inst.CKLN_DELAY_OVR_VAL = 7'b0000000;
defparam mipi_dhpy_inst.D0LN_DELAY_EN = 1'b0;
defparam mipi_dhpy_inst.D0LN_DELAY_OVR_VAL = 7'b0000000;
defparam mipi_dhpy_inst.D0LN_DESKEW_BYPASS = 1'b0;
defparam mipi_dhpy_inst.D1LN_DELAY_EN = 1'b0;
defparam mipi_dhpy_inst.D1LN_DELAY_OVR_VAL = 7'b0000000;
defparam mipi_dhpy_inst.D1LN_DESKEW_BYPASS = 1'b0;
defparam mipi_dhpy_inst.D2LN_DELAY_EN = 1'b0;
defparam mipi_dhpy_inst.D2LN_DELAY_OVR_VAL = 7'b0000000;
defparam mipi_dhpy_inst.D2LN_DESKEW_BYPASS = 1'b0;
defparam mipi_dhpy_inst.D3LN_DELAY_EN = 1'b0;
defparam mipi_dhpy_inst.D3LN_DELAY_OVR_VAL = 7'b0000000;
defparam mipi_dhpy_inst.D3LN_DESKEW_BYPASS = 1'b0;
defparam mipi_dhpy_inst.DESKEW_EN_LOW_DELAY = 1'b0;
defparam mipi_dhpy_inst.DESKEW_EN_ONE_EDGE = 1'b0;
defparam mipi_dhpy_inst.DESKEW_FAST_LOOP_TIME = 4'b0000;
defparam mipi_dhpy_inst.DESKEW_FAST_MODE = 1'b0;
defparam mipi_dhpy_inst.DESKEW_HALF_OPENING = 6'b000000;
defparam mipi_dhpy_inst.DESKEW_LSB_MODE = 2'b00;
defparam mipi_dhpy_inst.DESKEW_M = 3'b000;
defparam mipi_dhpy_inst.DESKEW_M_TH = 13'b0000000000000;
defparam mipi_dhpy_inst.DESKEW_MAX_SETTING = 7'b0000000;
defparam mipi_dhpy_inst.DESKEW_ONE_CLK_EDGE_EN = 1'b0;
defparam mipi_dhpy_inst.DESKEW_RST_BYPASS = 1'b0;
defparam mipi_dhpy_inst.RX_ALIGN_BYTE = 8'b10111000;
defparam mipi_dhpy_inst.RX_BYTE_LITTLE_ENDIAN = 1'b1;
defparam mipi_dhpy_inst.RX_CLK_1X_SYNC_SEL = 1'b0;

```



```

defparam mipi_dhpy_inst.RX_HS_8BIT_MODE = 1'b0;
defparam mipi_dhpy_inst.RX_INVERT = 1'b0;
defparam mipi_dhpy_inst.RX_LANE_ALIGN_EN = 1'b0;
defparam mipi_dhpy_inst.RX_ONE_BYTE0_MATCH = 1'b0;
defparam mipi_dhpy_inst.RX_RD_START_DEPTH = 5'b00001;
defparam mipi_dhpy_inst.RX_SYNC_MODE = 1'b0;
defparam mipi_dhpy_inst.RX_WORD_ALIGN_BYPASS = 1'b0;
defparam mipi_dhpy_inst.RX_WORD_ALIGN_DATA_VLD_SRC_SEL
= 1'b0;
defparam mipi_dhpy_inst.RX_WORD_LITTLE_ENDIAN = 1'b1;
defparam mipi_dhpy_inst.TX_BYPASS_MODE = 1'b0;
defparam mipi_dhpy_inst.TX_BYTECLK_SYNC_MODE = 1'b0;
defparam mipi_dhpy_inst.TX_HS_8BIT_MODE = 1'b0;
defparam mipi_dhpy_inst.TX_OCLK_USE_CIBCLK = 1'b0;
defparam mipi_dhpy_inst.TX_RD_START_DEPTH = 5'b00001;
defparam mipi_dhpy_inst.TX_SYNC_MODE = 1'b0;
defparam mipi_dhpy_inst.TX_WORD_LITTLE_ENDIAN = 1'b1;
defparam mipi_dhpy_inst.EQ_CS_LANE0 = 3'b100;
defparam mipi_dhpy_inst.EQ_CS_LANE1 = 3'b100;
defparam mipi_dhpy_inst.EQ_CS_LANE2 = 3'b100;
defparam mipi_dhpy_inst.EQ_CS_LANE3 = 3'b100;
defparam mipi_dhpy_inst.EQ_CS_LANECK = 3'b100;
defparam mipi_dhpy_inst.EQ_RS_LANE0 = 3'b100;
defparam mipi_dhpy_inst.EQ_RS_LANE1 = 3'b100;
defparam mipi_dhpy_inst.EQ_RS_LANE2 = 3'b100;
defparam mipi_dhpy_inst.EQ_RS_LANE3 = 3'b100;
defparam mipi_dhpy_inst.EQ_RS_LANECK = 3'b100;
defparam mipi_dhpy_inst.HSCLK_LANE_LN0 = 1'b1;
defparam mipi_dhpy_inst.HSCLK_LANE_LN1 = 1'b1;
defparam mipi_dhpy_inst.HSCLK_LANE_LN2 = 1'b1;
defparam mipi_dhpy_inst.HSCLK_LANE_LN3 = 1'b1;
defparam mipi_dhpy_inst.HSCLK_LANE_LNCK = 1'b0;
defparam mipi_dhpy_inst.HSREG_EN_LN0 = 1'b0;
defparam mipi_dhpy_inst.HSREG_EN_LN1 = 1'b0;
defparam mipi_dhpy_inst.HSREG_EN_LN2 = 1'b0;
defparam mipi_dhpy_inst.HSREG_EN_LN3 = 1'b0;
defparam mipi_dhpy_inst.HSREG_EN_LNCK = 1'b0;
defparam mipi_dhpy_inst.LANE_DIV_SEL = 2'b00;
defparam mipi_dhpy_inst.ALP_ED_EN_LANE0 = 1'b1;
defparam mipi_dhpy_inst.ALP_ED_EN_LANE1 = 1'b1;
defparam mipi_dhpy_inst.ALP_ED_EN_LANE2 = 1'b1;
defparam mipi_dhpy_inst.ALP_ED_EN_LANE3 = 1'b1;
defparam mipi_dhpy_inst.ALP_ED_EN_LANECK = 1'b1;
defparam mipi_dhpy_inst.ALP_ED_TST_LANE0 = 1'b0;

```

```
defparam mipi_dhpy_inst.ALP_ED_TST_LANE1 = 1'b0;
defparam mipi_dhpy_inst.ALP_ED_TST_LANE2 = 1'b0;
defparam mipi_dhpy_inst.ALP_ED_TST_LANE3 = 1'b0;
defparam mipi_dhpy_inst.ALP_ED_TST_LANECK = 1'b0;
defparam mipi_dhpy_inst.ALP_EN_LN0 = 1'b0;
defparam mipi_dhpy_inst.ALP_EN_LN1 = 1'b0;
defparam mipi_dhpy_inst.ALP_EN_LN2 = 1'b0;
defparam mipi_dhpy_inst.ALP_EN_LN3 = 1'b0;
defparam mipi_dhpy_inst.ALP_EN_LNCK = 1'b0;
defparam mipi_dhpy_inst.ALP_HYS_EN_LANE0 = 1'b1;
defparam mipi_dhpy_inst.ALP_HYS_EN_LANE1 = 1'b1;
defparam mipi_dhpy_inst.ALP_HYS_EN_LANE2 = 1'b1;
defparam mipi_dhpy_inst.ALP_HYS_EN_LANE3 = 1'b1;
defparam mipi_dhpy_inst.ALP_HYS_EN_LANECK = 1'b1;
defparam mipi_dhpy_inst.ALP_TH_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.ALP_TH_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.ALP_TH_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.ALP_TH_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.ALP_TH_LANECK = 4'b1000;
defparam mipi_dhpy_inst.ANA_BYTECLK_PH = 2'b00;
defparam mipi_dhpy_inst.BIT_REVERSE_LN0 = 1'b0;
defparam mipi_dhpy_inst.BIT_REVERSE_LN1 = 1'b0;
defparam mipi_dhpy_inst.BIT_REVERSE_LN2 = 1'b0;
defparam mipi_dhpy_inst.BIT_REVERSE_LN3 = 1'b0;
defparam mipi_dhpy_inst.BIT_REVERSE_LNCK = 1'b0;
defparam mipi_dhpy_inst.BYPASS_TXHCLKEN = 1'b1;
defparam mipi_dhpy_inst.BYPASS_TXHCLKEN_SYNC = 1'b0;
defparam mipi_dhpy_inst.BYTE_CLK_POLAR = 1'b0;
defparam mipi_dhpy_inst.BYTE_REVERSE_LN0 = 1'b0;
defparam mipi_dhpy_inst.BYTE_REVERSE_LN1 = 1'b0;
defparam mipi_dhpy_inst.BYTE_REVERSE_LN2 = 1'b0;
defparam mipi_dhpy_inst.BYTE_REVERSE_LN3 = 1'b0;
defparam mipi_dhpy_inst.BYTE_REVERSE_LNCK = 1'b0;
defparam mipi_dhpy_inst.EN_CLKB1X = 1'b1;
defparam mipi_dhpy_inst.EQ_PBIAS_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.EQ_PBIAS_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.EQ_PBIAS_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.EQ_PBIAS_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.EQ_PBIAS_LANECK = 4'b1000;
defparam mipi_dhpy_inst.EQ_ZLD_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.EQ_ZLD_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.EQ_ZLD_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.EQ_ZLD_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.EQ_ZLD_LANECK = 4'b1000;
```

```
defparam mipi_dhpy_inst.HIGH_BW_LANE0 = 1'b1;
defparam mipi_dhpy_inst.HIGH_BW_LANE1 = 1'b1;
defparam mipi_dhpy_inst.HIGH_BW_LANE2 = 1'b1;
defparam mipi_dhpy_inst.HIGH_BW_LANE3 = 1'b1;
defparam mipi_dhpy_inst.HIGH_BW_LANECK = 1'b1;
defparam mipi_dhpy_inst.HSREG_VREF_CTL = 3'b100;
defparam mipi_dhpy_inst.HSREG_VREF_EN = 1'b0;
defparam mipi_dhpy_inst.HSRX_DLY_CTL_CK = 7'b0000000;
defparam mipi_dhpy_inst.HSRX_DLY_CTL_LANE0 = 7'b0000000;
defparam mipi_dhpy_inst.HSRX_DLY_CTL_LANE1 = 7'b0000000;
defparam mipi_dhpy_inst.HSRX_DLY_CTL_LANE2 = 7'b0000000;
defparam mipi_dhpy_inst.HSRX_DLY_CTL_LANE3 = 7'b0000000;
defparam mipi_dhpy_inst.HSRX_DLY_SEL_LANE0 = 1'b0;
defparam mipi_dhpy_inst.HSRX_DLY_SEL_LANE1 = 1'b0;
defparam mipi_dhpy_inst.HSRX_DLY_SEL_LANE2 = 1'b0;
defparam mipi_dhpy_inst.HSRX_DLY_SEL_LANE3 = 1'b0;
defparam mipi_dhpy_inst.HSRX_DLY_SEL_LANECK = 1'b0;
defparam mipi_dhpy_inst.HSRX_DUTY_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.HSRX_DUTY_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.HSRX_DUTY_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.HSRX_DUTY_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.HSRX_DUTY_LANECK = 4'b1000;
defparam mipi_dhpy_inst.HSRX_EN = 1'b1;
defparam mipi_dhpy_inst.HSRX_EQ_EN_LANE0 = 1'b1;
defparam mipi_dhpy_inst.HSRX_EQ_EN_LANE1 = 1'b1;
defparam mipi_dhpy_inst.HSRX_EQ_EN_LANE2 = 1'b1;
defparam mipi_dhpy_inst.HSRX_EQ_EN_LANE3 = 1'b1;
defparam mipi_dhpy_inst.HSRX_EQ_EN_LANECK = 1'b1;
defparam mipi_dhpy_inst.HSRX_IBIAS = 4'b0011;
defparam mipi_dhpy_inst.HSRX_IBIAS_TEST_EN = 1'b0;
defparam mipi_dhpy_inst.HSRX_IMARG_EN = 1'b1;
defparam mipi_dhpy_inst.HSRX_LANESEL = 4'b1111;
defparam mipi_dhpy_inst.HSRX_LANESEL_CK = 1'b1;
defparam mipi_dhpy_inst.HSRX_ODT_EN = 1'b1;
defparam mipi_dhpy_inst.HSRX_ODT_TST = 4'b0000;
defparam mipi_dhpy_inst.HSRX_ODT_TST_CK = 1'b0;
defparam mipi_dhpy_inst.HSRX_SEL = 4'b0000;
defparam mipi_dhpy_inst.HSRX_STOP_EN = 1'b0;
defparam mipi_dhpy_inst.HSRX_TST = 4'b0000;
defparam mipi_dhpy_inst.HSRX_TST_CK = 1'b0;
defparam mipi_dhpy_inst.HSRX_WAIT4EDGE = 1'b1;
defparam mipi_dhpy_inst.HSTX_EN_LN0 = 1'b0;
defparam mipi_dhpy_inst.HSTX_EN_LN1 = 1'b0;
defparam mipi_dhpy_inst.HSTX_EN_LN2 = 1'b0;
```

```
defparam mipi_dhpy_inst.HSTX_EN_LN3 = 1'b0;
defparam mipi_dhpy_inst.HSTX_EN_LNCK = 1'b0;
defparam mipi_dhpy_inst.HYST_NCTL = 2'b01;
defparam mipi_dhpy_inst.HYST_PCTL = 2'b01;
defparam mipi_dhpy_inst.IBIAS_TEST_EN = 1'b0;
defparam mipi_dhpy_inst.LB_CH_SEL = 1'b0;
defparam mipi_dhpy_inst.LB_EN_LN0 = 1'b0;
defparam mipi_dhpy_inst.LB_EN_LN1 = 1'b0;
defparam mipi_dhpy_inst.LB_EN_LN2 = 1'b0;
defparam mipi_dhpy_inst.LB_EN_LN3 = 1'b0;
defparam mipi_dhpy_inst.LB_EN_LNCK = 1'b0;
defparam mipi_dhpy_inst.LB_POLAR_LN0 = 1'b0;
defparam mipi_dhpy_inst.LB_POLAR_LN1 = 1'b0;
defparam mipi_dhpy_inst.LB_POLAR_LN2 = 1'b0;
defparam mipi_dhpy_inst.LB_POLAR_LN3 = 1'b0;
defparam mipi_dhpy_inst.LB_POLAR_LNCK = 1'b0;
defparam mipi_dhpy_inst.LOW_LPRX_VTH = 1'b0;
defparam mipi_dhpy_inst.LPBK_DATA2TO1 = 4'b0000;
defparam mipi_dhpy_inst.LPBK_DATA2TO1_CK = 1'b0;
defparam mipi_dhpy_inst.LPBK_EN = 1'b0;
defparam mipi_dhpy_inst.LPBK_SEL = 4'b0000;
defparam mipi_dhpy_inst.LPBKTST_EN = 4'b0000;
defparam mipi_dhpy_inst.LPBKTST_EN_CK = 1'b0;
defparam mipi_dhpy_inst.LPRX_EN = 1'b1;
defparam mipi_dhpy_inst.LPRX_TST = 4'b0000;
defparam mipi_dhpy_inst.LPRX_TST_CK = 1'b0;
defparam mipi_dhpy_inst.LPTX_DAT_POLAR_LN0 = 1'b0;
defparam mipi_dhpy_inst.LPTX_DAT_POLAR_LN1 = 1'b0;
defparam mipi_dhpy_inst.LPTX_DAT_POLAR_LN2 = 1'b0;
defparam mipi_dhpy_inst.LPTX_DAT_POLAR_LN3 = 1'b0;
defparam mipi_dhpy_inst.LPTX_DAT_POLAR_LNCK = 1'b0;
defparam mipi_dhpy_inst.LPTX_EN_LN0 = 1'b1;
defparam mipi_dhpy_inst.LPTX_EN_LN1 = 1'b1;
defparam mipi_dhpy_inst.LPTX_EN_LN2 = 1'b1;
defparam mipi_dhpy_inst.LPTX_EN_LN3 = 1'b1;
defparam mipi_dhpy_inst.LPTX_EN_LNCK = 1'b1;
defparam mipi_dhpy_inst.LPTX_NIMP_LN0 = 3'b100;
defparam mipi_dhpy_inst.LPTX_NIMP_LN1 = 3'b100;
defparam mipi_dhpy_inst.LPTX_NIMP_LN2 = 3'b100;
defparam mipi_dhpy_inst.LPTX_NIMP_LN3 = 3'b100;
defparam mipi_dhpy_inst.LPTX_NIMP_LNCK = 3'b100;
defparam mipi_dhpy_inst.LPTX_PIMP_LN0 = 3'b100;
defparam mipi_dhpy_inst.LPTX_PIMP_LN1 = 3'b100;
defparam mipi_dhpy_inst.LPTX_PIMP_LN2 = 3'b100;
```

```
defparam mipi_dhpy_inst.LPTX_PIMP_LN3 = 3'b100;
defparam mipi_dhpy_inst.LPTX_PIMP_LNCK = 3'b100;
defparam mipi_dhpy_inst.MIPI_PMA_DIS_N = 1'b1;
defparam mipi_dhpy_inst.PGA_BIAS_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.PGA_BIAS_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.PGA_BIAS_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.PGA_BIAS_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.PGA_BIAS_LANECK = 4'b1000;
defparam mipi_dhpy_inst.PGA_GAIN_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.PGA_GAIN_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.PGA_GAIN_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.PGA_GAIN_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.PGA_GAIN_LANECK = 4'b1000;
defparam mipi_dhpy_inst.RX_ODT_TRIM_LANE0 = 4'b1000;
defparam mipi_dhpy_inst.RX_ODT_TRIM_LANE1 = 4'b1000;
defparam mipi_dhpy_inst.RX_ODT_TRIM_LANE2 = 4'b1000;
defparam mipi_dhpy_inst.RX_ODT_TRIM_LANE3 = 4'b1000;
defparam mipi_dhpy_inst.RX_ODT_TRIM_LANECK = 4'b1000;
defparam mipi_dhpy_inst.SLEWN_CTL_LN0 = 4'b1111;
defparam mipi_dhpy_inst.SLEWN_CTL_LN1 = 4'b1111;
defparam mipi_dhpy_inst.SLEWN_CTL_LN2 = 4'b1111;
defparam mipi_dhpy_inst.SLEWN_CTL_LN3 = 4'b1111;
defparam mipi_dhpy_inst.SLEWN_CTL_LNCK = 4'b1111;
defparam mipi_dhpy_inst.SLEWP_CTL_LN0 = 4'b1111;
defparam mipi_dhpy_inst.SLEWP_CTL_LN1 = 4'b1111;
defparam mipi_dhpy_inst.SLEWP_CTL_LN2 = 4'b1111;
defparam mipi_dhpy_inst.SLEWP_CTL_LN3 = 4'b1111;
defparam mipi_dhpy_inst.SLEWP_CTL_LNCK = 4'b1111;
defparam mipi_dhpy_inst.STP_UNIT = 2'b11;
defparam mipi_dhpy_inst.TERMN_CTL_LN0 = 4'b1000;
defparam mipi_dhpy_inst.TERMN_CTL_LN1 = 4'b1000;
defparam mipi_dhpy_inst.TERMN_CTL_LN2 = 4'b1000;
defparam mipi_dhpy_inst.TERMN_CTL_LN3 = 4'b1000;
defparam mipi_dhpy_inst.TERMN_CTL_LNCK = 4'b1000;
defparam mipi_dhpy_inst.TERMP_CTL_LN0 = 4'b1000;
defparam mipi_dhpy_inst.TERMP_CTL_LN1 = 4'b1000;
defparam mipi_dhpy_inst.TERMP_CTL_LN2 = 4'b1000;
defparam mipi_dhpy_inst.TERMP_CTL_LN3 = 4'b1000;
defparam mipi_dhpy_inst.TERMP_CTL_LNCK = 4'b1000;
defparam mipi_dhpy_inst.TEST_EN_LN0 = 1'b0;
defparam mipi_dhpy_inst.TEST_EN_LN1 = 1'b0;
defparam mipi_dhpy_inst.TEST_EN_LN2 = 1'b0;
defparam mipi_dhpy_inst.TEST_EN_LN3 = 1'b0;
defparam mipi_dhpy_inst.TEST_EN_LNCK = 1'b0;
```

```
defparam mipi_dhpy_inst.TEST_N_IMP_LN0 = 1'b0;  
defparam mipi_dhpy_inst.TEST_N_IMP_LN1 = 1'b0;  
defparam mipi_dhpy_inst.TEST_N_IMP_LN2 = 1'b0;  
defparam mipi_dhpy_inst.TEST_N_IMP_LN3 = 1'b0;  
defparam mipi_dhpy_inst.TEST_N_IMP_LNCK = 1'b0;  
defparam mipi_dhpy_inst.TEST_P_IMP_LN0 = 1'b0;  
defparam mipi_dhpy_inst.TEST_P_IMP_LN1 = 1'b0;  
defparam mipi_dhpy_inst.TEST_P_IMP_LN2 = 1'b0;  
defparam mipi_dhpy_inst.TEST_P_IMP_LN3 = 1'b0;  
defparam mipi_dhpy_inst.TEST_P_IMP_LNCK = 1'b0;  
defparam mipi_dhpy_inst.TXDP_EN_LNCK = 1'b1;  
defparam mipi_dhpy_inst.TXDP_EN_LN1 = 1'b1;  
defparam mipi_dhpy_inst.TXDP_EN_LN2 = 1'b1;  
defparam mipi_dhpy_inst.TXDP_EN_LN3 = 1'b1;  
defparam mipi_dhpy_inst.TXDP_EN_LNCK = 1'b1;
```

3.2 MIPI D-PHY RX

3.2.1 Port Description

Table 3-3 MIPI D-PHY RX Ports

Port	I/O	Description
MIPI INTERFACE Signals		
CK_N.	inout	CK Lane Completion Input
CK_P.	inout	CK Lane True Input
DX<0~3>_N	inout	Data Lane <0~3> Completion Input
DX<0~3>_P	inout	Data Lane <0~3> True Input
RESET and CLOCK Signals		
RESET	input	Reset signal: 1'b1: reset all;
PWRON	input	Power-On Control: 1'b1 - HSRX on 1'b0 - HSRX off to standby in low power state
HSRX_STOP	input	HSRX Clock Stop Signal for synchronization
DRST_N.	input	Digital reset, active low
RX_CLK_1X	input	1X clock from fabric, max 93.75MHz@1.5Gbps
RX_CLK_O.	output	HSRX Clock output
CONFIG Signals		
RX_INVERT	input	Data density selection
HS_8BIT_MODE	input	Selection of data width to Fabric
BYTE_LENDIAN	input	Bit data Little/Big-endian of 8bit
World_LENDIAN	input	Data little/big endian of dual bytes. Not used in 8bit data output mode
FIFO_RD_STD[2:0]	input	FIFO read threshold. Can only be 1 in 8bit mode
WALIGN_by	input	Word aligner bypass
ONE_BYTE0_MATCH	input	Byte count match in word aligner
LALIGN_en	input	Lane aligner enable
WALIGN_DVLD	input	Word aligner input data valid from the fabric
HSRX Signals		
D<0~3>LN_HSRXD[7/15:0]	output	Data Lane <0~3> HS data output to fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16

Port	I/O	Description
D<0~3>LN_HSRXD_VLD	output	Data Lane <0~3> HS data output valid to fabric
HSRX_en_CK	input	CK Lane HSRX enabled
D<0~3>LN_HSRX_DREN	input	Data Lane <0~3> HSRX driver enabled
HSRX_ODTEN_CK	input	CK Lane HSRX ODT enabled
HSRX_ODTEN_D<0~3>	input	Data Lane <0~3> HSRX ODT enabled
LPRX Signals		
LPRX_EN_D<0~3>	input	Data Lane <0~3> LPRX enabled
Di_LPRX<0~3>_N.	output	Data Lane <0~3> Completion Pad LPRX input
Di_LPRX<0~3>_P.	output	Data Lane <0~3> True Pad LPRX input
LPRX_en_CK	input	CK Lane LPRX enabled
Di_LPRXCK_n	output	CK Lane Completion Pad LPRX input
Di_LPRXCK_P.	output	CK Lane True Pad LPRX input
LPTX Signals		
LPTX_EN_D<0~3>	input	Data Lane <0~3> LPTX enabled
Do_LPTX<0~3>_n	input	Data Lane <0~3> Completion Pad LPTX output
Do_LPTX<0~3>_P.	input	Data Lane <0~3> True Pad LPTX output
LPTX_en_CK	input	CK Lane LPTX enabled
Do_LPTXCK_n	input	CK Lane Completion Pad LPTX output
Do_LPTXCK_P.	input	CK Lane True Pad LPTX output
DE-SKEW Signals		
DESKEW_by	input	
DESKEW_EN_OEDGE	input	
DESKEW_LSB_MODE[2:0]	input	
DESKEW_M[2:0]	input	Counter threshold for confirming edge
DESKEW_MSET[6:0]	input	
DESKEW_MTH[12:0]	input	Counter threshold for searching one edge
DESKEW_LNSEL[2:0]	input	Selection of lane to config delay overwrite value and make lane in de-skew delay overwrite mode
DESKEW_HALF_OPENING[5:0]	input	
DESKEW_OCLKEDG_en	input	Select one clock edge (pos-edge/neg-edge) to calculate the de-skew delay
DESKEW_OWVAL[6:0]	input	
DESKEW_REQ	input	De-skew function request to all data lanes
DESKEW_ERROR	output	4 data lane de-skew result error report

Port	I/O	Description
D<0~3>LN_DESKEW_DONE	output	Lane <0~3> de-skew done
HSRX_DLYDIR_LANE<0~3>	input	Data Lane <0~3> Direction for HSRX De-skew Delay Control: 0 Count Up; 1 Count Down
HSRX_DLYDIR_CK	input	CK Lane Direction for HSRX De-skew Delay Control: 0 Count Up; 1 Count Down
HSRX_DLYLDN_LANE<0~3>	input	Data Lane <0~3>: Load HSRX De-skew Delay Control input from Fuse: 1'b0 load
HSRX_DLYLDN_CK	input	CK Lane: Load HSRX De-skew Delay Control input from Fuse: 1'b0 load
HSRX_DLYMV_LANE<0~3>	input	Data Lane <0~3>: Enable HSRX De-skew Delay Control to count: 1'b1 move
HSRX_DLYMV_CK	input	CK Lane: Enable HSRX De-skew Delay Control to count: 1'b1 move
EQCS_LANE[2:0]	input	Data Lane <0~3>: Equalizer Source degeneration capacitor setting, 3b'000 smallest Cap; 3b'111 biggest Cap, default 3'b100
EQCS_CK[2:0]	input	CK Lane: Equalizer Source degeneration capacitor setting, 3b'000 smallest Cap; 3b'111 biggest Cap, default 3'b100

4 IP Generation

For ease of use, the Gowin Software offers IP packaging of the hard-core MIPI D-PHY. The packaged IP is suitable for most of the usage scenarios. If there is a further need, it can be instantiated in a primitive manner under the guidance of technical support.

4.1 MIPI D-PHY TX IP

4.1.1 Port Description

Table 4-1 MIPI D-PHY TX IP Ports

Port	I/O	Description
MIPI Physical Layer Signal ¹		
Ck_n	inout	clock lane, differential clock signal n-terminal
ck_p	inout	clock lane, differential clock signal p-terminal
d<0~3>_n	inout	data lane< 0~ 3>, differential data signal n-terminal
d<0~3>_p	inout	data lane< 0~ 3>, differential data signal p-terminal
Clock and Reset signal		
tx_drst_n	input	TX digital reset, active-low.
clkin ²	input	PLL Input Clock
clkout ⁴	output	PLL output clock clkout ⁴ .
tx_clk_o ²	output	Byte Clock, the primary operating clock; The HS TX signal is mostly synchronized to this clock.
HS TX Signal		
txhclk_en	input	High Speed Clock enable, active-high
txdpen_inck	input	Clock Lane driver enable, active-high.
txdpen_in<0~3>	input	Data Lane driver enable, active-high.

Port	I/O	Description
hstxen_inck	input	Clock Lane HS transmit enable, active-high.
hstxen_in<0~3>	input	Data Lane <0~3> HS transmit enable, active-high.
hstxd_vld	input	HS TX data valid indicator, active-high.
ckln_hstxd[7/15:0]	input	Clock Channel HS Transmit Data Input; 8 bit width is selected for 8-bits mode and 16 bit width for 16-bits mode; Typically 8'b0101_0101 or 16'h0101_0101_0101_0101 is used as a fixed input.
d<0~3>ln_hstxd[7/15:0]	input	Data channel <0~3> HS Transmit Data Input; 8 bit width is selected for 8-bits mode and 16 bit width for 16-bits mode; With a bit width of 16, the lower 8 bits are sent to the line first.
LP TX Signal		
lptxen_inck	input	Clock Lane LP transmit enable, active-high.
do_lptxck_n	input	Clock lane LP transmit data input, n-terminal.
do_lptxck_p	input	Clock lane LP transmit data input, p-terminal.
lptxen_in<0~3>	input	Data lane <0~3> LP transmit enable, active-high.
do_lptx<0~3>_n	input	Data lane < 0~ 3> LP transmit data input, n-terminal.
do_lptx<0~3>_p	input	Data lane < 0~ 3> LP transmit data input, p-terminal.
LP RX Signal		
lprx_en_ck	input	Clock lane LP receive enable, active-high.
di_lprxck_n	output	Clock lane LP receive data output, n-terminal.
di_lprxck_p	output	Clock lane LP receive data output, p-terminal.
lprx_en_d<0~3>	input	Data lane <0~3> LP receive enable, active-high.
di_lprx<0~3>_n	output	Data lane < 0~ 3> LP receive data output, n-terminal.
di_lprx<0~3>_p	output	Data lane < 0~ 3> LP Receive data output, p-terminal.

Note!

- The physical interface is connected directly to the MIPI-proprietary pins (see "[UG985E, GW5A-25 Pinout](#)") and no connection is required for instantiation; D-PHY RX shares the MIPI-proprietary pin with D-PHY TX.
- The MIPI D-PHY TX IP internally instantiates a D-PHY TX-specific PLL (see Section 5.1 in "[UG286E, Arora V Clock User Guide](#)." for PLL resources) to produce the required clock. D-PHY TX uses the VCO of this PLL as the reference clock source; byte clock is introduced by tx_clk_o at:

$$f_{tx_clk_o} = \langle DPHY \text{ Data Rate} \rangle / \langle 8 \text{ or } 16 \rangle;$$

The user-defined clock is introduced by clkout4 at the following frequencies:

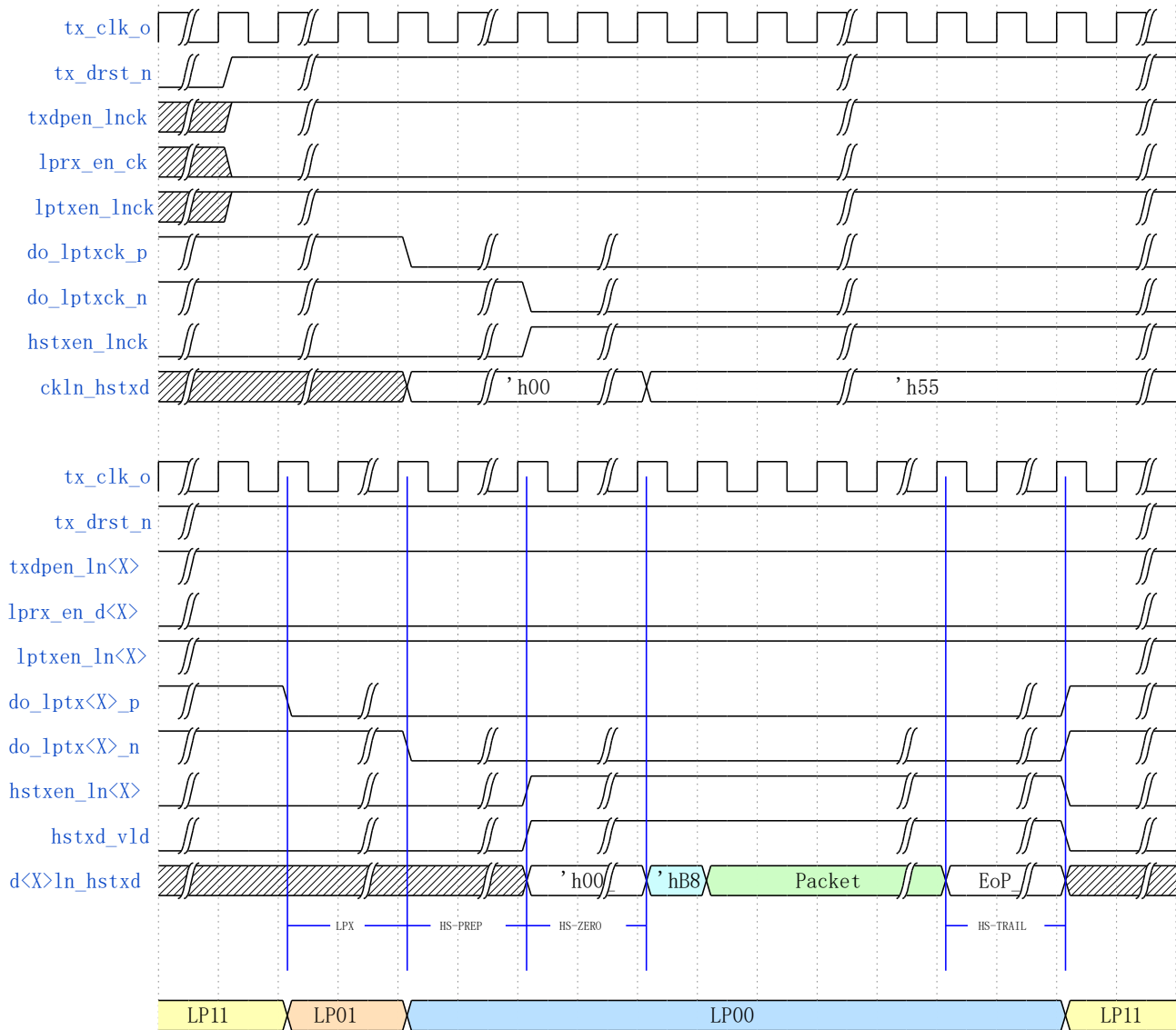
$$f_{clkout4} = f_{VCO} / \langle CLKOUT4 \text{ Divider Factor} \rangle.$$

Note!

f_{VCO} is not necessarily equal to the D-PHY Data Rate.

4.1.2 Typical Timing

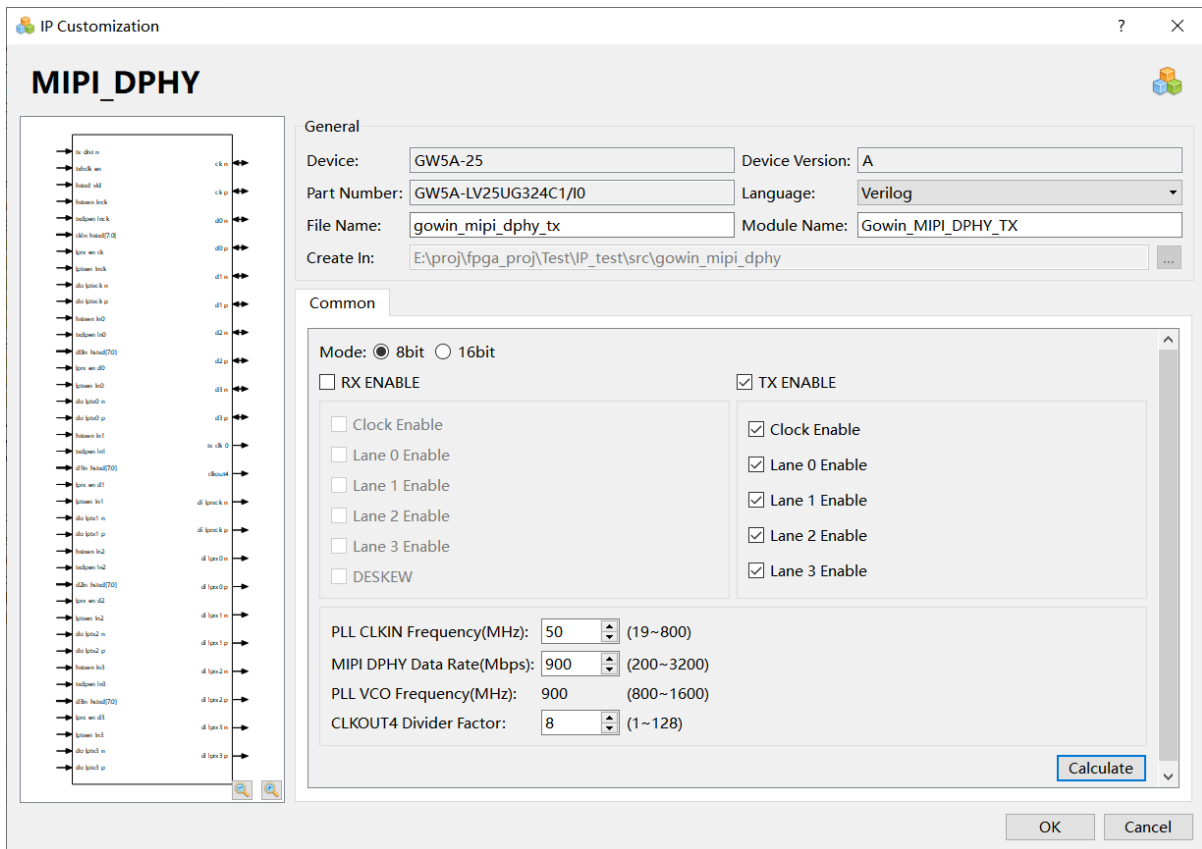
Figure 4-1 MIPI D-PHY TX Typical Timing



<X>=0, 1, 2 or 3

4.1.3 Configuration Description

Figure 4-2 MIPI D- PHY Configuration (TX)



1. General Configuration

The General configuration is used to configure the information about the generated IP design file.

- Device: Display information about the configured Device.
- Part Number: Display the configured Part Number.
- Language: Hardware description language used to generate the IP design files. Click the drop-down list to select the language, including Verilog and VHDL .
- Module Name: The module name of the generated IP design files. It can be re-edited in the text box on the right. Module name cannot be the same as the primitive name. If it is the same, an error will be reported.
- File Name: The file name of the generated IP design files. It can be re-edited in the text box on the right.
- Create In: The path in which the generated IP files will be stored. Enter the target path in the right box or select the target path using the Select button on the right side of the text box.

2. Common Configuration

- Mode: Select 8-bits or 16-bits mode;
- RX_ENABLE: Select MIPI D-PHY RX to enable TX configuration;
- Clock Lane Enable: Disable or enable clock lane.
- Lane 0 Enable: Disable or enable lane 0.
- Lane 1 Enable: Disable or enable lane 1.
- Lane 2 Enable: Disable or enable lane 2.
- Lane 3 Enable: Disable or enable lane 3.

3. TX Exclusive PLL Configuration

- PLL CLKIN Frequency: PLL input clock frequency (19~800 MHz);
- MIPI D-PHY Data Rate: MIPI single lane rate (200~3200 Mbps);
- PLL VCO Frequency: PLL VCO frequency prompt (800~1600 MHz), click the "Calculate" button to update the prompt;
- CLKOUT4 Divider Factor: PLL output frequency division factor (1-128), the final output frequency of the clkout4 signal is the VCO frequency divided by this frequency division factor.

4. Port Diagram

The port diagram shows the results of the current IP Core configuration, as shown on the left of Figure 4-2.

4.2 MIPI D-PHY RX IP

4.2.1 Port Description

Table 4-2 MIPI D-PHY RX IP Ports

Port	I/O	Description
MIPI physical layer Signal ³		
Ck_n	inout	clock lane, differential clock signal n-terminal
ck_p	inout	clock lane, differential clock signal p-terminal
d<0~3>_n	inout	data lane< 0~ 3>, differential data signal n-terminal
d<0~3>_p	inout	data lane< 0~ 3>, differential data signal p-terminal
Clock and Reset signal		
rx_drst_n	input	RX digital reset, active-low.
rx_clk_o ⁴	output	Byte Clock, the primary operating clock; The HS RX signal is mostly synchronized to this clock.
HS RX Signal		
hsrx_en_ck	input	Clock lane HS receive enable, active-high.
hsrx_en_d<0~3>	input	Data lane <0~3> HS receive enable, active-high.
hsrx_odten_ck	input	Clock lane terminal resistance enable, active-high.
hsrx_odten_d<0~3>	input	Data lane <0~3> terminal resistance enable, active-high.
d<0~3>ln_hsrxd_vld	output	Data lane <0~3> HS data valid indication, active-high.
d<0~3>ln_hsrxd[7/15:0]	output	Data lane <0~3> HS data output; Bit width can be selected from 8bit or 16bit.
LP RX Signal		
lprx_en_ck	input	Clock lane LP receive enable, active-high.
di_lprxck_n	output	Clock lane LP receive data output, n-terminal.
di_lprxck_p	output	Clock lane LP receive data output, p-terminal.
lprx_en_d<0~3>	input	Data lane <0~3> LP receive enable, active-high.
di_lprx<0~3>_n	output	Data lane < 0~ 3> LP receive data output, n-terminal.
di_lprx<0~3>_p	output	Data lane < 0~ 3> LP Receive data output, p-terminal.
LP TX Signal		
lptxen_inck	input	Clock Lane LP transmit enable, active-high.
do_lptxck_n	input	Clock lane LP transmit data input, n-terminal.
do_lptxck_p	input	Clock lane LP transmit data input, p-terminal.
lptxen_in<0~3>	input	Data lane <0~3> LP transmit enable, active-high.

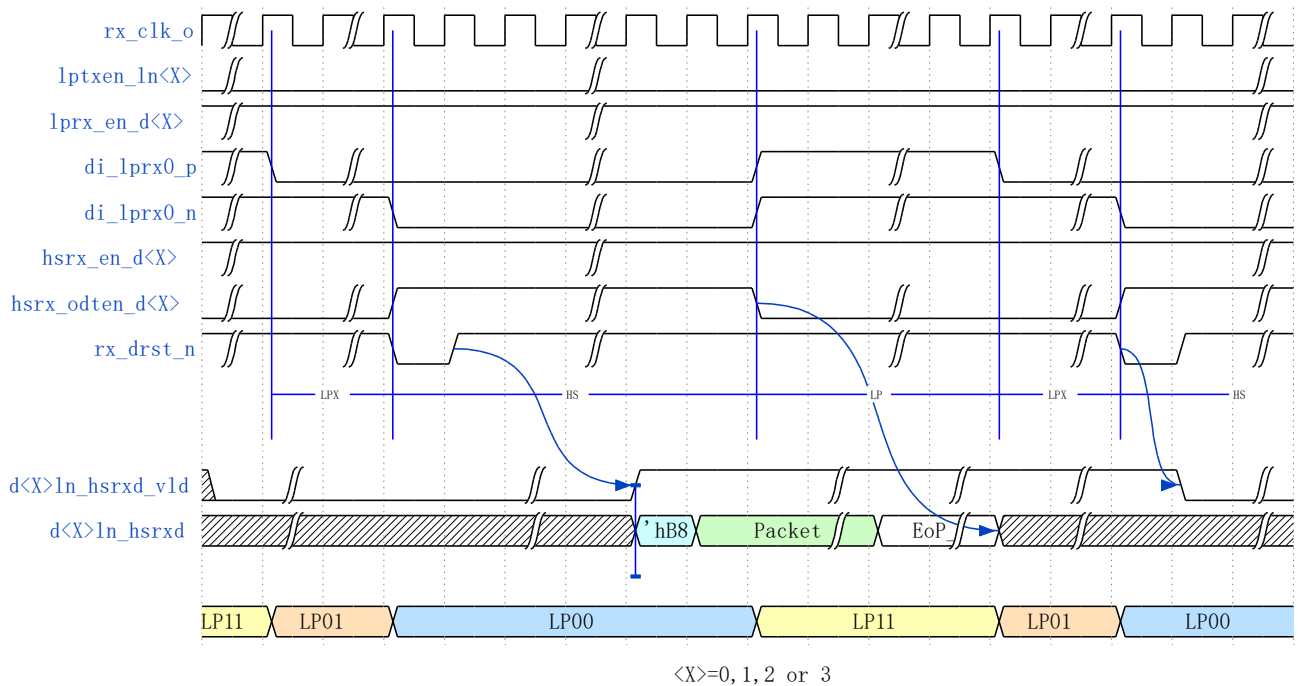
Port	I/O	Description
do_lptx<0~3>_n	input	Data lane < 0~ 3> LP transmit data input, n-terminal.
do_lptx<0~3>_p	input	Data lane < 0~ 3> LP transmit data input, p-terminal.
De-skew signal⁵		
d<0~3>ln_deskew_req	input	Data channel <0~3> de-skew request.
D<0~3>ln_deskew_done	output	Data channel <0~3> de-skew completion indication.
d<0~3>ln_deskew_error	output	Data channel <0~3> de-skew error indication.

Note!

- The physical interface is connected directly to the MIPI-proprietary pins (see [UG985, GW5A-25 Device Pinout Manual](#)) and no connection is required during the instantiation; D-PHY RX shares the MIPI-proprietary pins with D-PHY TX.
- The byte clock is introduced by rx_clk_o and normally has a frequency of: $f_{rx_clk_o} = \langle DPHY \text{ Data Rate} \rangle / \langle 8 \text{ or } 16 \rangle$.
- These signals are only present when the DESKEW option is enabled.

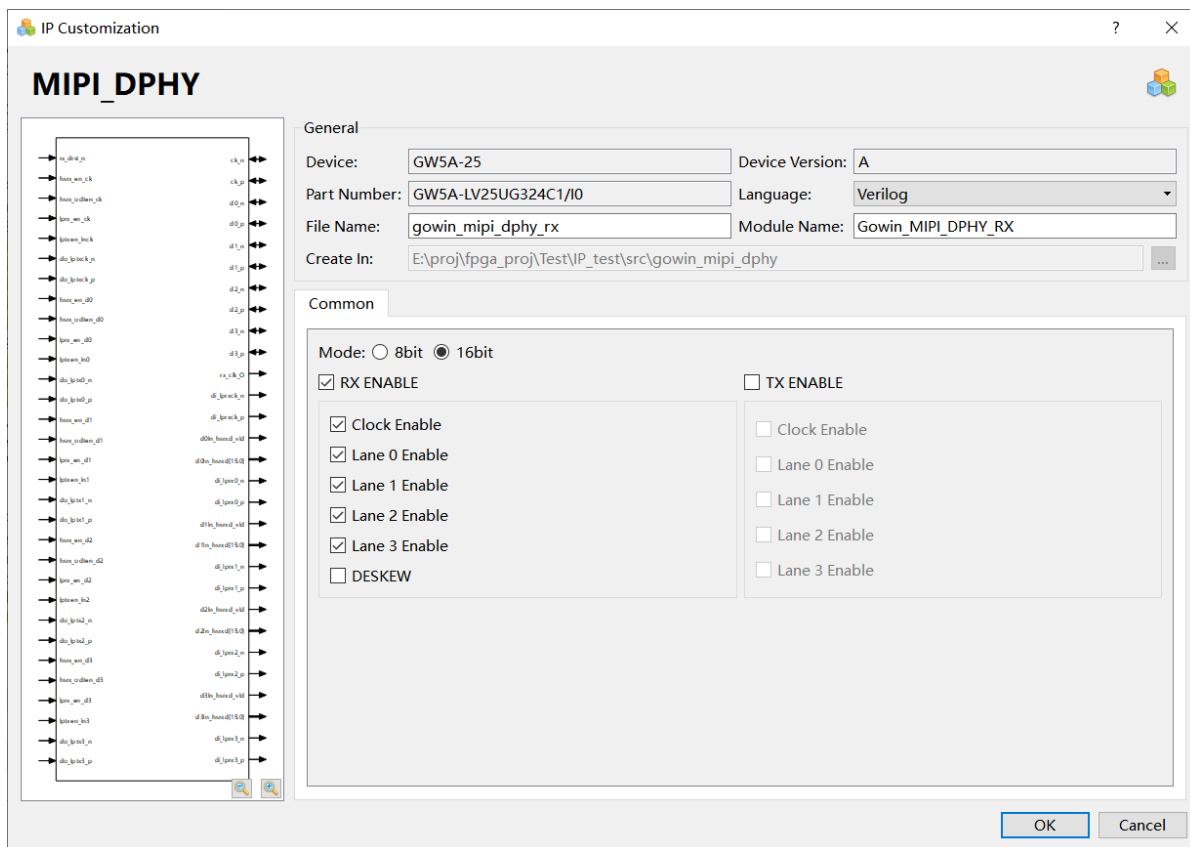
4.2.2 Typical Timing

Figure 4-3 MIPI D-PHY RX Typical Timing



4.2.3 Configuration Description

Figure 4-4 MIPI D- PHY Configuration (RX)



1. General Configuration

The General configuration is used to configure the information about the generated IP design file.

- Device: Display information about the configured Device.
- Part Number: Display the configured Part Number.
- Language: Hardware description language used to generate the IP design files. Click the drop-down list to select the language, including Verilog and VHDL .
- Module Name: Configure the top module name of the generated IP. It can be re-edited in the text box on the right. Module name cannot be the same as the primitive name. If it is the same, an error will be reported.
- File Name: The file name of the generated IP design files. It can be re-edited in the text box on the right.
- Create In: The path in which the generated IP files will be stored. Enter the target path in the right box or select the target path using the Select button on the right side of the text box.

2. Common Configuration

- Mode: Select 8-bits or 16-bits mode;
- RX_ENABLE: Select MIPI D-PHY RX to enable RX configuration;
- Clock Lane Enable: Disable or enable clock lane.
- Lane 0 Enable: Disable or enable lane 0.
- Lane 1 Enable: Disable or enable lane 1.
- Lane 2 Enable: Disable or enable lane 2.
- Lane 3 Enable: Disable or enable lane 3.
- DESKEW: Enables or disables the de-skew signal.

3. Port Diagram

The port diagram shows the results of the current IP Core configuration, as shown on the left of Figure 4-4.

Appendix **A** MIPI D-PHY Rate Table

Table A-1 MIPI D-PHY Rates (Arora Family)

Resolution	Frame Rate (HZ)	Bits Per Pixel (Bits)	Total Data Rate (Mbps)	Lane	Per Lane Bit Rate (Mbps)	Recommended Gearing Ratio (1:N)	Per Lane Fabric Clock (MHz)
FHD 1920x1080p (2200x1125)	60	8	1188	2	594.0	8	74.25
		10	1485	2	742.5	8	92.81
		16	2376	2	1188.0	8	148.50
		18	2673	4	668.3	8	83.53
		24	3564	4	891.0	8	111.38
	120	8	2376	2	1188.0	8	148.50
		10	2970	4	742.5	8	92.81
		16	4752	4	1188.0	8	148.50
		18	5346	8	668.3	8	83.53
		24	7128	8	891.0	8	111.38
UHD 3840x2160p (4400x2250)	30	8	2376	4	594.0	8	74.25
		10	2970	4	742.5	8	92.81
		16	4752	4	1188.0	8	148.50
		18	5346	8	668.3	8	83.53
		24	7128	8	891.0	8	111.38
	60	8	4752	4	1188.0	8	148.50
		10	5940	8	742.5	8	92.81
		16	9504	8	1188.0	8	148.50

