



Arora V Hardened MIPI D-PHY ユーザーガイド

UG296-1.0J, 2023-09-28

著作権について(2023)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

GOWIN高云、Gowin、及びGOWINSEMIは、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

免責事項

当社は、GOWINSEMI Terms and Conditions of Sale(GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的に拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

バージョン履歴

日付	バージョン	説明
2023/09/28	1.0J	初版。

目次

目次	iv
図一覧	vi
表一覧	vii
1 本マニュアルについて	1
1.1 マニュアル内容	1
1.2 関連ドキュメント	1
1.3 用語、略語	2
1.4 テクニカル・サポートとフィードバック	2
2 概要	3
2.1 特徴	3
2.2 機能の説明	4
2.2.1 MIPI D-PHY RX	4
2.2.2 MIPI D-PHY TX	5
3 MIPI D-PHY プリミティブ	6
3.1 MIPI D-PHY	6
3.1.1 ポートの説明	6
3.1.2 パラメータの説明	13
3.1.3 プリミティブのインスタンス化	14
3.2 MIPI D-PHY RX	24
3.2.1 ポートの説明	24
4 IP の呼び出し	27
4.1 MIPI D-PHY TX IP	27
4.1.1 ポートの説明	27
4.1.2 典型的なタイミング	30
4.1.3 構成の説明	31

4.2 MIPI D-PHY RX IP.....	33
4.2.1 ポートの説明	33
4.2.2 典型的なタイミング.....	35
4.2.3 構成の説明	36
付録 A MIPI D-PHY のデータレート表.....	38

図一覧

図 2-1 MIPI D-PHY RX の構造	4
図 2-2 MIPI D-PHY TX の構造	5
図 4-1 MIPI D-PHY TX の典型的なタイミング	30
図 4-2 MIPI D-PHY の構成画面 (TX)	31
図 4-3 MIPI D-PHY RX の典型的なタイミング	35
図 4-4 MIPI D-PHY の構成画面(RX)	36

表一覧

表 1-1 用語、略語	2
表 3-1 MIPI D-PHY のポートの説明	6
表 3-2 MIPI D-PHY のパラメータの説明	13
表 3-3 MIPI D-PHY RX のポートの説明.....	24
表 4-1 MIPI D-PHY TX IP のトップレベルのポート	27
表 4-2 MIPI D-PHY RX IP のトップレベルのポート.....	33
表 A-1 MIPI D-PHY データレート(Arora ファミリー)	39

1 本マニュアルについて

1.1 マニュアル内容

本マニュアルは、ユーザーが Arora V の MIPI D-PHY を使いこなせるよう、その機能、ポート、呼び出しと構成などについて説明しています。

1.2 関連ドキュメント

GOWIN セミコンダクターのホームページ www.gowinsemi.com/ja から、以下の関連ドキュメントがダウンロード、参考できます：

- GW5AT シリーズ FPGA 製品データシート([DS981](#))
- GW5A シリーズ FPGA 製品データシート([DS1103](#))
- GW5AST シリーズ FPGA 製品データシート([DS1104](#))
- GW5AS-138 FPGA 製品データシート([DS1114](#))
- GW5AR シリーズ FPGA 製品データシート([DS1108](#))
- GW5A シリーズ FPGA 製品(オートモーティブ)データシート([DS1113](#))
- GW5AT シリーズ FPGA 製品(オートモーティブ)データシート([DS1111](#))

1.3 用語、略語

表 1-1 に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
CSI	Camera Serial Interface	カメラ・シリアル・インターフェース
DSI	Display Serial Interface	ディスプレイ・シリアル・インターフェース
HS	High Speed	高速
I/O	Input/output	入力/出力
LP	Low Power	低消費電力
MIPI	Mobile Industry Processor Interface	モバイル産業用プロセッサ・インターフェース

1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

ホームページ : www.gowinsemi.com/ja

E-mail : support@gowinsemi.com

2 概要

Arora V FPGA には、『MIPI Alliance Standard for D-PHY Specification』バージョン 2.1 をサポートするハードコア MIPI D-PHY RX および MIPI D-PHY TX^[1]があります。このハードコア MIPI D-PHY は、DSI インターフェースおよび CSI-2 インターフェースに適しています。

注記：

[1]現在、Gowin の 138K デバイスは MIPI D-PHY RX のみ、25K デバイスは MIPI D-PHY RX と MIPI D-PHY TX をサポートしています。

2.1 特徴

MIPI D-PHY の主な特徴は以下に示す通りです：

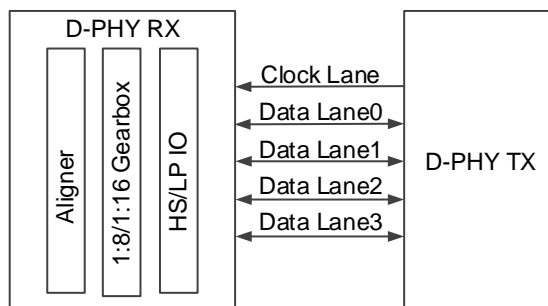
- 単方向の高速(HS、High-speed)モードをサポート。データレーンあたりの最大転送レートは 2.5Gbps。単一クアッドは最大 10Gbps の転送レート(4 データレーン)をサポート。単一チップは最大 20 Gbps の転送レート(8 データレーン)をサポート。
- 最大 2 クアッドの MIPI D-PHY をサポート。各クアッドは最大 4 つのデータレーンおよび 1 つのクロックレーンを提供。
- 双方向の低消費電力(LP、Low-power)モードをサポート。転送レートは 10Mbps。
- 高速同期、Word Alignment と Lane Alignment をサポート。
- MIPI D-PHY RX が 1:8 モード及び 1:16 モードをサポート。
- MIPI D-PHY TX が 8:1 モード及び 16:1 モードをサポート。
- MIPI DSI および MIPI CSI-2 リンク層をサポート。
- 専用の MIPI Bank あり。
- GW5AT-138 デバイスは 2 クアッドの MIPI D-PHY RX をサポート。
- GW5A-25 デバイスは 1 クアッドの MIPI D-PHY(RX/TX に構成可能)をサポート。

2.2 機能の説明

2.2.1 MIPI D-PHY RX

MIPI D-PHY RX には、HS/LP IO、1:8/1:16 Gearbox、および Aligner が含まれています(図 2-1)。

図 2-1 MIPI D-PHY RX の構造



HS/LP I/O

ODT の動的切り替えがサポートされ、LP TX/RX と HS RX の動的切り替えがサポートされます。

1:8/1:16 Gearbox

8 ビットまたは 16 ビットのモードがサポートされます。プリミティブでは HS_8BIT_MODE パラメータにより設定可能です。IP では Mode オプションにより設定可能です。

Aligner

Word align(ワードアライン)および Lane align(レーンアライン)がサポートされます。MIPI_DPHY_RX ポートの WALIGN_BY および LALIGN_EN により設定可能です。そのうち、Word align のキーは、ユーザ一定義 (ALIGN_BYTE)をサポートし、MIPI_DPHY_RX の ONE_BYTE0_MATCH ポートにより 2 バイトまたは 3 バイトモードとして設定することをサポートします。

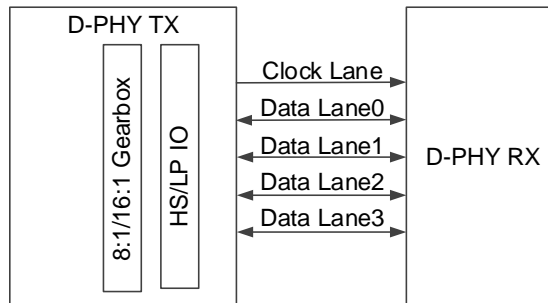
注記：

MIPI_DPHY_RX のポートの説明については、3 MIPI D-PHY プリミティブ > 3.1.1 ポートの説明、または 4.2 MIPI D-PHY RX IP > 4.2.1 ポートの説明を参照してください。

2.2.2 MIPI D-PHY TX

MIPI D-PHY TX には、HS/LP IO、8:1/16:1 Gearbox が含まれています(図 2-1)。

図 2-2 MIPI D-PHY TX の構造



HS/LP I/O

LP TX/RX と HS TX の動的切り替えがサポートされます。

8:1/16:1 Gearbox

8 ビットまたは 16 ビットのモードがサポートされます。
HS_8BIT_MODE パラメータにより設定可能です。

注記：

MIPI_DPHY_TX のポートの説明については、3 MIPI D-PHY プリミティブ > ポートの説明を参照してください。

3 MIPI D-PHY プリミティブ

3.1 MIPI D-PHY

3.1.1 ポートの説明

表 3-1 MIPI D-PHY のポートの説明

ポート	I/O	説明
MIPI INTERFACE Signals		
CK_N	入出力	CK Lane Complement Input
CK_P	入出力	CK Lane True Input
D<0~3>_N	入出力	Data Lane <0~3> Complement Input
D<0~3>_P	入出力	Data Lane <0~3> True Input
RESET Signals		
RESET	入力	Reset signal: 1'b1: reset all;
RX_DRST_N	入力	RX digital reset, active low
TX_DRST_N	入力	TX digital reset, active low
PWRON_RX	入力	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	入力	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	入力	HSRX Clock Stop Signal for synchronization
WALIGN_DVLD	入力	word aligner input data valid from the fabric
CLOCK Signals		
CK0	入力	HSTX: ck0

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

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

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

ポート	I/O	説明
		1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D0LN_HSTXD	入力	data lane0 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D1LN_HSTXD	入力	data lane1 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D2LN_HSTXD	入力	data lane2 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
D3LN_HSTXD	入力	data lane3 HS data input from fabric 1:8 Mode: Data Width=8 1:16Mode: Data Width=16
HSTXD_VLD	入力	HS_TX Data Valid input from fabric
HSTXEN_LNCK	入力	CK Lane0: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN0	入力	Data Lane0: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN1	入力	Data Lane1: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN2	入力	Data Lane2: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
HSTXEN_LN3	入力	Data Lane3: <ul style="list-style-type: none"> ● 1'b1: HSTX enabled ● 1'b0: HSTX disabled
TXDPEN_LN0	入力	txdpen_ln0, 1'b1 enabled
TXDPEN_LN1	入力	txdpen_ln1, 1'b1 enabled
TXDPEN_LN2	入力	txdpen_ln2, 1'b1 enabled
TXDPEN_LN3	入力	txdpen_ln3, 1'b1 enabled
TXDPEN_LNCK	入力	txdpen_lnc, 1'b1 enabled
TXHCLK_EN	入力	txhclk_en, 1'b1 enabled

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

ポート	I/O	説明
DO_LPTXCK_P	入力	CK Lane True Pad LPTX output
LPTXEN_LNCK	入力	CK Lane: 1'b1 LPTX enabled
LPTXEN_LN0	入力	Data Lane0: 1'b1 LPTX enabled
LPTXEN_LN1	入力	Data Lane1: 1'b1 LPTX enabled
LPTXEN_LN2	入力	Data Lane2: 1'b1 LPTX enabled
LPTXEN_LN3	入力	Data Lane3: 1'b1 LPTX enabled
ALP Signals		
ALPEDO_LANE0	出力	ALP Mode: Data Lane0 output
ALPEDO_LANE1	出力	ALP Mode: Data Lane1 output
ALPEDO_LANE2	出力	ALP Mode: Data Lane2 output
ALPEDO_LANE3	出力	ALP Mode: Data Lane3 output
ALPEDO_LANECK	出力	ALP Mode: CK Lane output
ALP_EDEN_LANE0	入力	ALP Mode: alp_eden_lane0
ALP_EDEN_LANE1	入力	ALP Mode: alp_eden_lane1
ALP_EDEN_LANE2	入力	ALP Mode: alp_eden_lane2
ALP_EDEN_LANE3	入力	ALP Mode: alp_eden_lane3
ALP_EDEN_LANECK	入力	ALP Mode: alp_eden_laneck
ALPEN_LN0	入力	ALP Mode:1'b1, Lane0 enabled
ALPEN_LN1	入力	ALP Mode:1'b1, Lane1 enabled
ALPEN_LN2	入力	ALP Mode:1'b1, Lane2 enabled
ALPEN_LN3	入力	ALP Mode:1'b1, Lane3 enabled
ALPEN_LNCK	入力	ALP Mode: 1'b1, CK lane enabled
MRDATA [7:0]	出力	mrdata
MA_INC	入力	ma_inc
MCLK	入力	mclk
MOPCODE	入力	mopcode
MWDATA [7:0]	入力	mwdata

3.1.2 パラメータの説明

表 3-2 MIPI D-PHY のパラメータの説明

パラメータ	デフォルト	説明
RX_ALIGN_BYTE	8'b10111000	KEY for word aligner and lane aligner
RX_BYTE_LITTLE_ENDIAN	1'b1	1'b1: Little endian
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 polarity 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_WORD_ALIGN_BYPASS	1'b0	
RX_WORD_ALIGN_DATA_VLD_SRC_SEL	1'b0	
RX_WORD_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 プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

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(txdpenn0),
.TXDPEN_LN1(txdpenn1),
.TXDPEN_LN2(txdpenn2),
.TXDPEN_LN3(txdpenn3),
.TXDPEN_LNCK(txdpennck),
.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'b00000000000000;
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_LANESSEL = 4'b1111;
defparam mipi_dhpy_inst.HSRX_LANESSEL_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_LN0 = 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 ポートの説明

表 3-3 MIPI D-PHY RX のポートの説明

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

ポート	I/O	説明
D<0~3>LN_HSRXD_VLD	出力	Data Lane <0~3> HS data output valid to fabric
HSRX_EN_CK	入力	CK Lane HSRX enabled
D<0~3>LN_HSRX_DREN	入力	Data Lane <0~3> HSRX driver enabled
HSRX_ODTEN_CK	入力	CK Lane HSRX ODT enabled
HSRX_ODTEN_D<0~3>	入力	Data Lane <0~3> HSRX ODT enabled
LPRX Signals		
LPRX_EN_D<0~3>	入力	Data Lane <0~3> LPRX enabled
DI_LPRX<0~3>_N	出力	Data Lane <0~3> Complement Pad LPRX input
DI_LPRX<0~3>_P	出力	Data Lane <0~3> True Pad LPRX input
LPRX_EN_CK	入力	CK Lane LPRX enabled
DI_LPRXCK_N	出力	CK Lane Complement Pad LPRX input
DI_LPRXCK_P	出力	CK Lane True Pad LPRX input
LPTX Signals		
LPTX_EN_D<0~3>	入力	Data Lane <0~3> LPTX enabled
DO_LPTX<0~3>_N	入力	Data Lane <0~3> Complement Pad LPTX output
DO_LPTX<0~3>_P	入力	Data Lane <0~3> True Pad LPTX output
LPTX_EN_CK	入力	CK Lane LPTX enabled
DO_LPTXCK_N	入力	CK Lane Complement Pad LPTX output
DO_LPTXCK_P	入力	CK Lane True Pad LPTX output
DE-SKEW Signals		
DESKEW_BY	入力	
DESKEW_EN_OEDGE	入力	
DESKEW_LSB_MODE[2:0]	入力	
DESKEW_M[2:0]	入力	counter threshold for confirming edge
DESKEW_MSET[6:0]	入力	
DESKEW_MTH[12:0]	入力	counter threshold for searching one edge
DESKEW_LNSEL[2:0]	入力	selection of lane to config delay overwrite value and make lane in de-skew delay overwrite mode
DESKEW_HALF_OPENING[5:0]	入力	
DESKEW_OCLKEDG_EN	入力	select one clock edge (pos-edge/neg-edge) to calculate the de-skew delay
DESKEW_OWVAL[6:0]	入力	
DESKEW_REQ	入力	De-skew function request to all data lanes
DESKEW_ERROR	出力	4 data lane de-skew result error report

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

4 IP の呼び出し

使いやすさのために、GOWIN ソフトウェアは、ハード MIPI D-PHY コアを IP としてパッケージしています。パッケージされた IP はほとんどの使用シナリオに適していますが、テクニカル・サポート・スタッフの指導の下、プリミティブのインスタンス化を行うことも可能です。

4.1 MIPI D-PHY TX IP

4.1.1 ポートの説明

表 4-1 MIPI D-PHY TX IP のトップレベルのポート

ポート	I/O	説明
MIPI の物理インターフェース信号^[1]		
ck_n	入出力	N 側の差動クロック
ck_p	入出力	P 側の差動クロック
d<0~3>_n	入出力	N 側の差動データ<0~3>
d<0~3>_p	入出力	P 側の差動データ<0~3>
クロックとリセット		
tx_drst_n	入力	TX デジタル部のリセット、アクティブ Low
clkin ²	入力	PLL 入力クロック。
clkout4 ²	出力	PLL 出力クロック clkout4。
tx_clk_o ²	出力	バイトクロック (Byte Clock)。主な動作クロック。 HS TX 信号はほとんどこのクロックに同期します。
HS TX 信号		

ポート	I/O	説明
txhclk_en	入力	高速クロックのイネーブル、アクティブ High
txdpen_lnc	入力	クロックレーン driver のイネーブル、アクティブ High
txdpen_ln<0~3>	入力	データレーン<0~3> driver のイネーブル、アクティブ High
hstxen_lnc	入力	クロックレーン HS 送信のイネーブル、アクティブ High
hstxen_ln<0~3>	入力	データレーン<0~3> HS 送信のイネーブル、アクティブ High
hstxd_vld	入力	HS 送信データ有効の指示。アクティブ High。
ckln_hstxd[7/15:0]	入力	クロックレーン HS 送信データ入力。 8-bits モードの場合、幅は 8 ビットで、16-bits モードの場合、幅は 16 ビットです。 通常、8'b0101_0101 または 16'b0101_0101_0101_0101 が固定入力として使用されます。
d<0~3>ln_hstxd[7/15:0]	入力	データレーン<0~3> HS 送信データ入力。 8-bits モードの場合、幅は 8 ビットで、16-bits モードの場合、幅は 16 ビットです。 幅が 16 ビットの場合、下位 8 ビットが最初に送信されます。
LP TX 信号		
lptxen_lnc	入力	クロックレーン LP 送信のイネーブル、アクティブ High
do_lptxck_n	入力	クロックレーン LP 送信データ入力、N 側。
do_lptxck_p	入力	クロックレーン LP 送信データ入力、P 側。
lptxen_ln<0~3>	入力	データレーン<0~3> LP 送信のイネーブル、アクティブ High
do_lptx<0~3>_n	入力	データレーン<0~3> LP 送信データ入力、N 側。
do_lptx<0~3>_p	入力	データレーン<0~3> LP 送信データ入力、P 側。
LP RX 信号		
lprx_en_ck	入力	クロックレーン LP 受信のイネーブル、アクティブ High
di_lprxck_n	出力	クロックレーン LP 受信データ出力、N 側。
di_lprxck_p	出力	クロックレーン LP 受信データ出力、P 側。
lprx_en_d<0~3>	入力	データレーン<0~3> LP 受信のイネーブル、アクティブ High
di_lprx<0~3>_n	出力	データレーン<0~3> LP 受信データ出力、N 側。
di_lprx<0~3>_p	出力	データレーン<0~3> LP 受信データ出力、P 側。

注記：

- 物理インターフェースが MIPI 専用ピン(『GW5A-25 デバイス Pinout(UG985)』を参照)に直接接続されているため、 インスタンス際の接続は必要ありません。D-PHY RX と D-PHY TX は MIPI 専用ピンを共有します。
- MIPI D-PHY TX IP の内部では、必要なクロックの生成に使用される D-PHY TX 専用 PLL がインスタンス化されています(PLL リソースについては、『Arora V Clock ユーザーガイド(UG306)』を参照)。D-PHY TX はこの PLL の VCO をリファレンスクロック ソースとして使用します。バイトクロックは `tx_clk_o` から取得され、その周波数は：

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

ユーザー定義クロックは `clkout4` から取得され、その周波数は：

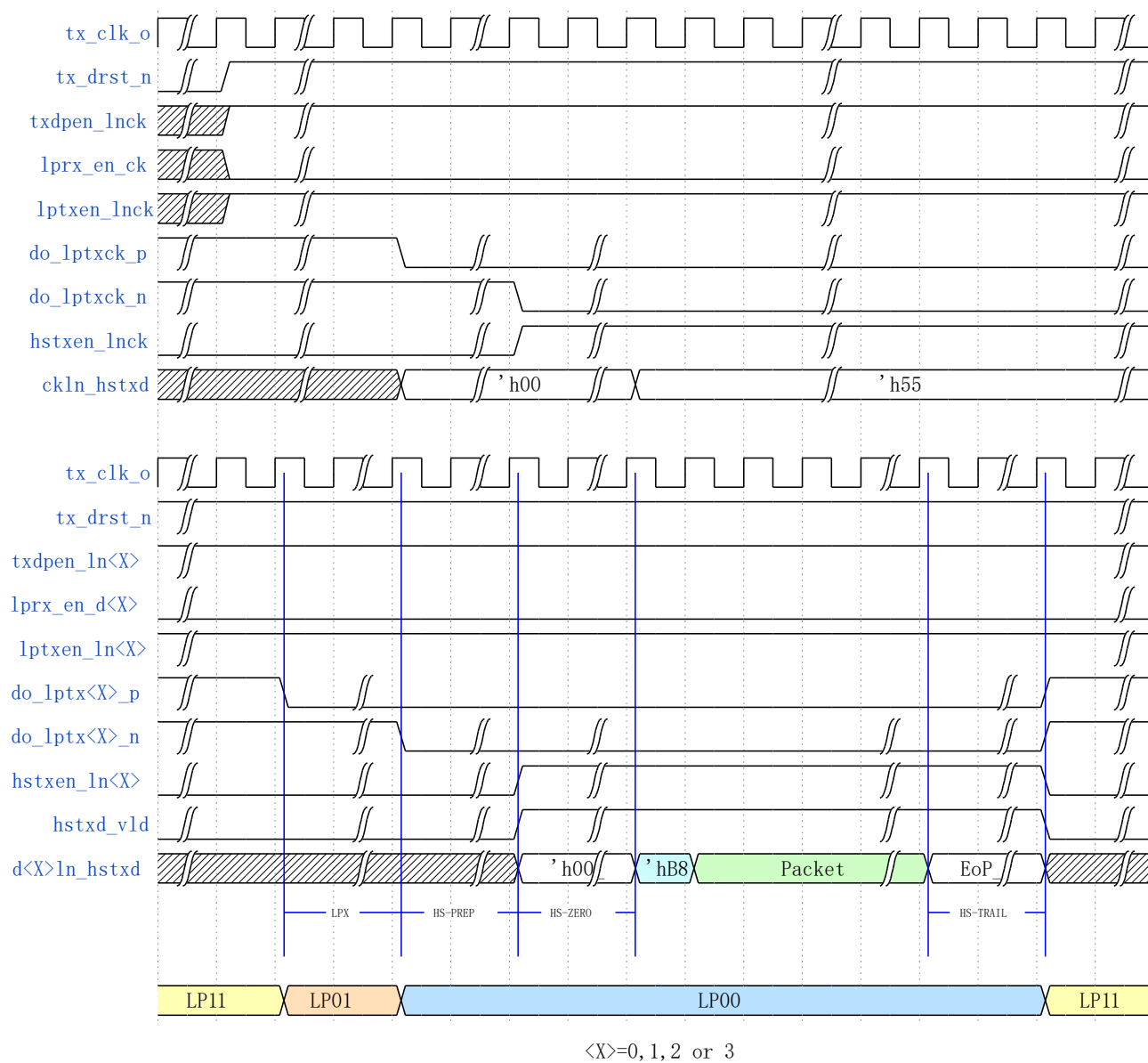
$$f_{clkout4} = f_{vco} / \langle CLKOUT4 \text{ Divider Factor} \rangle .$$

注記：

f_{vco} と D-PHY Data Rate は必ずしも等しいわけではありません。

4.1.2 典型的なタイミング

図 4-1 MIPI D-PHY TX の典型的なタイミング



4.1.3 構成の説明

図 4-2 MIPI D-PHY の構成画面 (TX)

The screenshot shows the 'MIPI_DPHY' IP Customization window. On the left is a pinout diagram. The right side has two tabs: 'General' and 'Common'. The 'General' tab contains the following fields:

- Device: GW5A-25
- Device Version: A
- Part Number: GW5A-LV25UG324C1/I0
- Language: Verilog
- File Name: gowin_mipi_dphy_tx
- Module Name: Gowin_MIPI_DPHY_TX
- Create In: E:\proj\ipga_proj\Test\IP_test\src\gowin_mipi_dphy

The 'Common' tab is partially visible, showing:

- Mode: 8bit (selected), 16bit
- RX ENABLE: unchecked
- TX ENABLE: checked
- Clock Enable: checked
- Lane 0 Enable: checked
- Lane 1 Enable: checked
- Lane 2 Enable: checked
- Lane 3 Enable: checked
- DESKEW: unchecked
- PLL CLKIN Frequency(MHz): 50 (19~800)
- MIPI DPHY Data Rate(Mbps): 900 (200~3200)
- PLL VCO Frequency(MHz): 900 (800~1600)
- CLKOUT4 Divider Factor: 8 (1~128)

Buttons for 'Calculate', 'OK', and 'Cancel' are at the bottom right.

1. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。

- **Device** : 対象デバイス。
- **Part Number** : 部品番号。
- **Language** : IP を実現するハードウェア記述言語。右側のドロップダウン・リストからターゲット言語(Verilog または VHDL)を選択します。
- **Module Name** : 生成される IP ファイルのモジュール名。右側のテキストボックスで編集できます。Module Name をプリミティブ名と同じにすることはできません。同じである場合、エラーが報告されます。
- **File Name** : 生成される IP ファイルのファイル名。右側のテキストボックスで再編集できます。
- **Create In** : 生成される IP ファイルのパス。右側のテキストボックスでパスを直接編集するか、テキストボックスの右側にある選択ボタンを使用してパスを選択できます。

2. Common タブ

- **Mode** : 8-bits モードまたは 16-bits モードを選択します。
- **TX_ENABLE** : MIPI D-PHY TX を有効にするかどうか選択します。
- **Clock Enable** : clock lane を有効にするかどうか選択します。
- **Lane 0 Enable** : data lane 0 を有効にするかどうか選択します。
- **Lane 1 Enable** : data lane 1 を有効にするかどうか選択します。
- **Lane 2 Enable** : data lane 2 を有効にするかどうか選択します。
- **Lane 3 Enable** : data lane 3 を有効にするかどうか選択します。

3. TX 専用 PLL の構成

- **PLL CLKIN Frequency** : PLL 入力クロックの周波数 (19~800 MHz)。
- **MIPI D-PHY Data Rate** : レーンあたりのデータレート (200~3200 Mbps)。
- **PLL VCO Frequency** : PLL VCO の周波数 (800 ~ 1600 MHz)。
"Calculate"ボタンをクリックするとこの周波数が更新されます。
- **CLKOUT4 Divider Factor** : PLL 出力の分周係数 (1~128)。
clkout4 信号の最終出力周波数は、VCO 周波数をこの分周係数で割ったものになります。

4. ポート図

ポート図に、IP Core の構成結果を示します(図 4-2)。

4.2 MIPI D-PHY RX IP

4.2.1 ポートの説明

表 4-2 MIPI D-PHY RX IP のトップレベルのポート

ポート	I/O	説明
MIPI の物理インターフェース信号^[1]		
ck_n	入出力	N 側の差動クロック
ck_p	入出力	P 側の差動クロック
d<0~3>_n	入出力	N 側の差動データ<0~3>
d<0~3>_p	入出力	P 側の差動データ<0~3>
クロックとリセット		
rx_drst_n	入力	RX デジタル部のリセット、アクティブ Low
rx_clk_o ^[4]	出力	バイトクロック (Byte Clock)。主な動作クロック。 HS RX 信号はほとんどこのクロックに同期します。
HS RX 信号		
hsrx_en_ck	入力	クロックレーン HS 受信のイネーブル、アクティブ High
hsrx_en_d<0~3>	入力	データレーン<0~3> HS 受信のイネーブル、アクティブ High
hsrx_odten_ck	入力	クロックレーン終端抵抗のイネーブル、アクティブ High
hsrx_odten_d<0~3>	入力	データレーン<0~3> 終端抵抗のイネーブル、アクティブ High
d<0~3>ln_hsrxd_vld	出力	データレーン<0~3> HS データ有効の指示、アクティブ High
d<0~3>ln_hsrxd[7/15:0]	出力	データレーン<0~3> HS データ出力。 8-bits モードの場合、幅は 8 ビットで、16-bits モードの場合、幅は 16 ビットです。
LP RX 信号		
lprx_en_ck	入力	クロックレーン LP 受信のイネーブル、アクティブ High
di_lprxck_n	出力	クロックレーン LP 受信データ出力、N 側。
di_lprxck_p	出力	クロックレーン LP 受信データ出力、P 側。
lprx_en_d<0~3>	入力	データレーン<0~3> LP 受信のイネーブル、アクティブ High

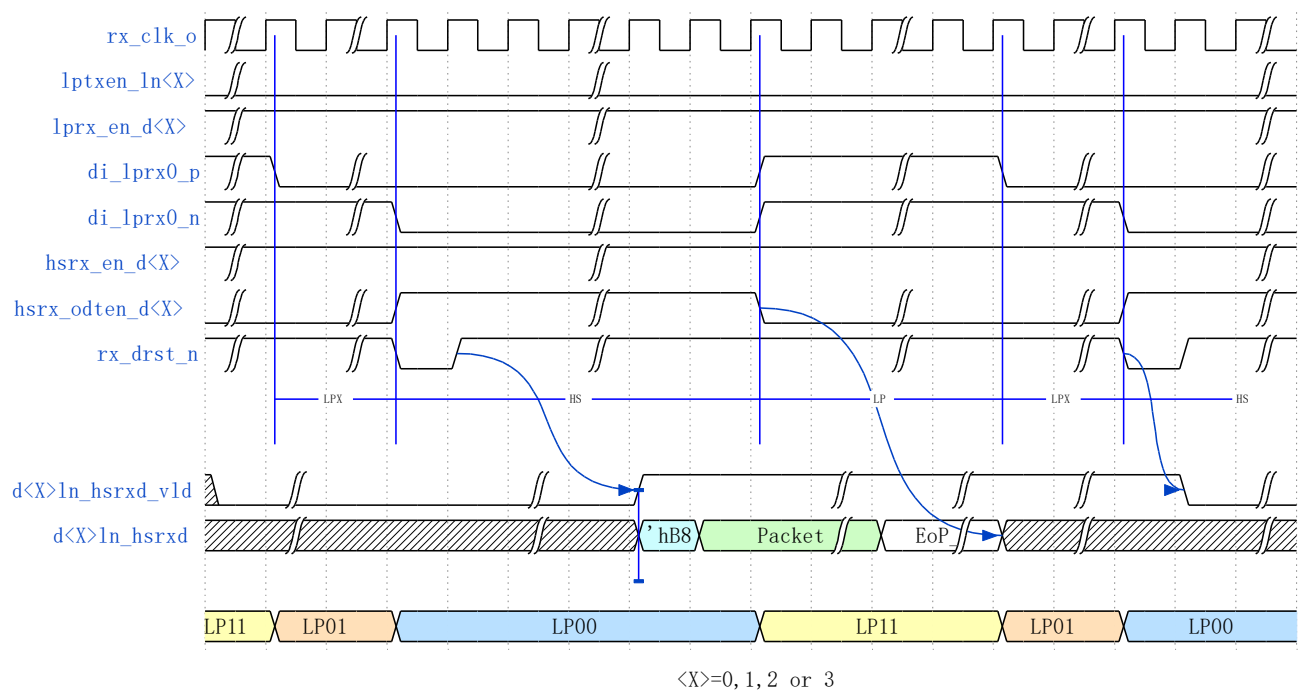
ポート	I/O	説明
di_lprx<0~3>_n	出力	データレーン<0~3> LP 受信データ出力、N 側。
di_lprx<0~3>_p	出力	データレーン<0~3> LP 受信データ出力、P 側。
LP TX 信号		
lptxen_inck	入力	クロックレーン LP 送信のイネーブル、アクティブ High
do_lptxck_n	入力	クロックレーン LP 送信データ入力、N 側。
do_lptxck_p	入力	クロックレーン LP 送信データ入力、P 側。
lptxen_in<0~3>	入力	データレーン<0~3> LP 送信のイネーブル、アクティブ High
do_lptx<0~3>_n	入力	データレーン<0~3> LP 送信データ入力、N 側。
do_lptx<0~3>_p	入力	データレーン<0~3> LP 送信データ入力、P 側。
de-skew 信号^[3]		
d<0~3>ln_deskew_req	入力	データレーン<0~3> de-skew 要求。
d<0~3>ln_deskew_done	出力	データレーン<0~3> de-skew 完了の指示。
d<0~3>ln_deskew_error	出力	データレーン<0~3> de-skew エラーの指示。

注記：

- 物理インターフェースが MIPI 専用ピン(『GW5A-25 デバイス Pinout([UG985](#))』を参照)に直接接続されているため、インスタンス際の接続は必要ありません。D-PHY RX と D-PHY TX は MIPI 専用ピンを共有します。
- バイトクロックは tx_clk_o から取得され、その周波数は通常：
 $f_{rx_clk_o} = <DPHY\ Data\ Rate> / <8\ or\ 16>.$
- これらの信号は、"DESKEW"オプションが有効な場合にのみ存在します。

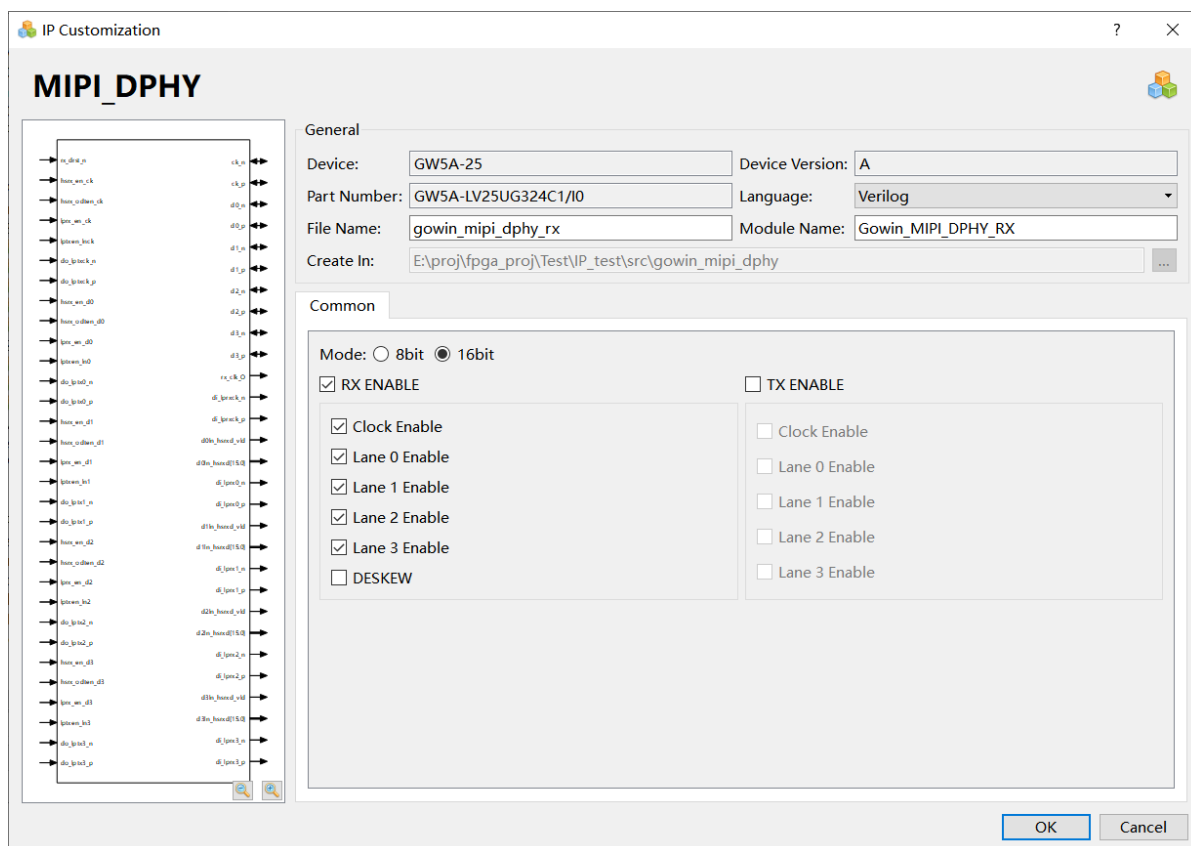
4.2.2 典型的なタイミング

図 4-3 MIPI D-PHY RX の典型的なタイミング



4.2.3 構成の説明

図 4-4 MIPI D-PHY の構成画面(RX)



1. General 構成タブ

General 構成タブは、IP ファイルの構成に使用されます。

- **Device** : 対象デバイス。
- **Part Number** : 部品番号。
- **Language** : IP を実現するハードウェア記述言語。右側のドロップダウン・リストからターゲット言語(Verilog または VHDL)を選択します。
- **Module Name** : 生成される IP のモジュール名右側のテキストボックスで編集できます。Module Name をプリミティブ名と同じにすることはできません。同じである場合、エラーが報告されます。
- **File Name** : 生成される IP ファイルのファイル名。右側のテキストボックスで再編集できます。
- **Create In** : 生成される IP ファイルのパス。右側のテキストボックスでパスを直接編集するか、テキストボックスの右側にある選択ボタンを使用してパスを選択できます。

2. Common タブ

- **Mode** : 8-bits モードまたは 16-bits モードを選択します。
- **RX_ENABLE** : MIPI D-PHY RX を有効にするかどうか選択します。
- **Clock Enable** : clock lane を有効にするかどうか選択します。
- **Lane 0 Enable** : data lane 0 を有効にするかどうか選択します。
- **Lane 1 Enable** : data lane 1 を有効にするかどうか選択します。
- **Lane 2 Enable** : data lane 2 を有効にするかどうか選択します。
- **Lane 3 Enable** : data lane 3 を有効にするかどうか選択します。
- **DESKEW** : de-skew 信号を有効にするかどうか選択します。

3. ポート図

ポート図に、IP Core の構成結果を示します(図 4-4)。

付録 **A** MIPI D-PHY のデータレート表

表 A-1 MIPI D-PHY データレート(Arora ファミリー)

Resolution	Frame Rate (HZ)	Bits Per Pixel (Bits)	Total Data Rate (Mbps)	Lane Number	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

