



XAPP1330 (v1.0) 2018 年 3 月 29 日

# High Speed SelectIO Wizard を使用した非同期データ キャプチャ

著者: Jim Tatsukawa

## 概要

このアプリケーション ノートおよび関連するリファレンス デザインは、High Speed SelectIO™ Wizard を使用して、UltraScale™ デバイスと UltraScale+™ デバイスでネイティブ モード I/O を使用して非同期レシーバーを生成する方法を示しています。リファレンス デザインは、『ネイティブ モードの高速 I/O インターフェイス』(XAPP1274) [参照 1] の非同期リファレンス デザインに基づいています。

3 チャンネルのトランスミッターとレシーバーのデザインは、KCU105 ボードなどのボード上に実装できます [参照 2]。リファレンス デザインは、High Speed SelectIO Wizard を使用してより簡単にピン配置が変更できるように修正されています。リファレンス デザインは KCU105 ボードをターゲットにしていますが、このアプリケーション ノートで説明されているガイドラインに従うことによって、ほかのボード用にデザインを変更できます。

このアプリケーション ノートのリファレンス デザイン ファイルは、ザイリンクスのウェブサイトからダウンロードできます。デザイン ファイルの詳細は、「非同期リファレンス デザイン」を参照してください。

## はじめに

非同期データ キャプチャにより、同期インターフェイスで一般に使用される転送クロックなしで、LVDS などの差動入力を使用してデータをキャプチャできます。代わりに、『ネイティブ高速 I/O インターフェイス』(XAPP1274) で説明されているとおり、基準クロックを使用してデータストリームからクロックが再キャプチャされます。クロック リカバリ アルゴリズムでは、RX\_BITSlice を直接制御することが必要です。これには High Speed SelectIO Wizard を設定して、クロック リカバリ アルゴリズムに接続する必要があります。

非同期データ キャプチャには次の制約があります。

- 1400Mb/s です。
- 入力は差動入力の必要があります。
- 差動入力に AC カップリングを使用する必要があります。

リファレンス デザインには次の制約があります。

- 移植する場合を考慮し、レシーバー (RX) とトランスミッター (TX) は異なるバイト グループに配置する必要があります。
- レシーバーは ASYNC モード DATA\_WIDTH = 4 を使用する必要があります。
- トランスミッターは DATA\_WIDTH = 8 を使用する必要があります。
- ピンを変更すると VHDL の変更が必要になる場合があります。
- RX および TX チャンネルは入力クロックを 625MHz 入力クロックとしています。
- 汎用 I/O の仮想 I/O (VIO) インターフェイスの制御には 125MHz の差動クロックが使用されます。

非同期データ キャプチャの動作に関する詳細は、『ネイティブ モードの高速 I/O インターフェイス』(XAPP1274) [参照 1] の「非同期データ キャプチャ インターフェイス」で説明されています。このアプリケーション ノートは、High Speed SelectIO Wizard から非同期デザインを作成する手法を説明しています。

このアプリケーション ノートは、次のトピックで構成されています。

- ・ 「非同期リファレンス デザイン」
- ・ 「シミュレーション」
- ・ 「ハードウェア動作」

## 非同期リファレンス デザイン

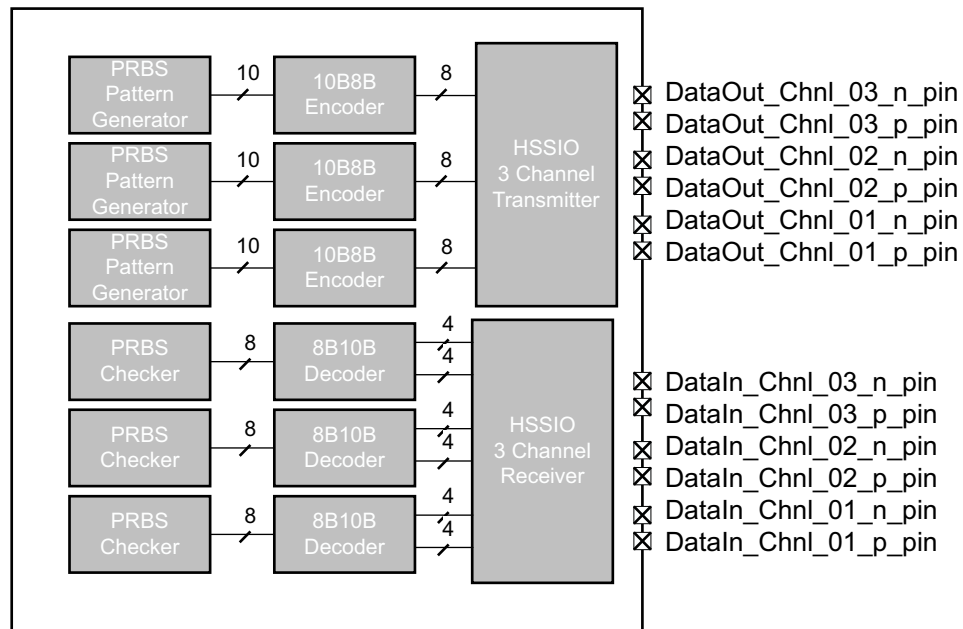
このアプリケーション ノートの [リファレンス デザイン ファイル](#) は、ザイリンクスのウェブサイトからダウンロードできます。

表 1 に、リファレンス デザインの詳細を示します。

表 1: リファレンス デザインの詳細

パラメーター	説明
<b>全般</b>	
開発者	Jim Tatsukawa, Marc Defossez
ターゲット デバイス	UltraScale FPGA
ソース コードの提供	あり
ソース コードの形式	VHDL および Verilog
既存のザイリンクス アプリケーション ノート/ リファレンス デザイン、またはサードパーティから デザインへのコード/IP の使用	あり
<b>シミュレーション</b>	
論理シミュレーションの実施	あり
タイミングシミュレーションの実施	あり
論理シミュレーションおよび タイミングシミュレーションでのテストベンチの利用	あり
テストベンチの形式	Verilog
使用したシミュレータ/バージョン	Vivado® シミュレータ
SPICE/IBIS シミュレーションの実施	なし
<b>インプリメンテーション</b>	
使用した合成ツール/バージョン	Vivado ツール 2017.4 またはそれ以降
使用したインプリメンテーション ツール/バージョン	Vivado ツール 2017.4 またはそれ以降
スタティック タイミング解析の実施	あり
<b>ハードウェア検証</b>	
ハードウェア検証の実施	あり
使用したハードウェア プラットフォーム	KCU105 ボード

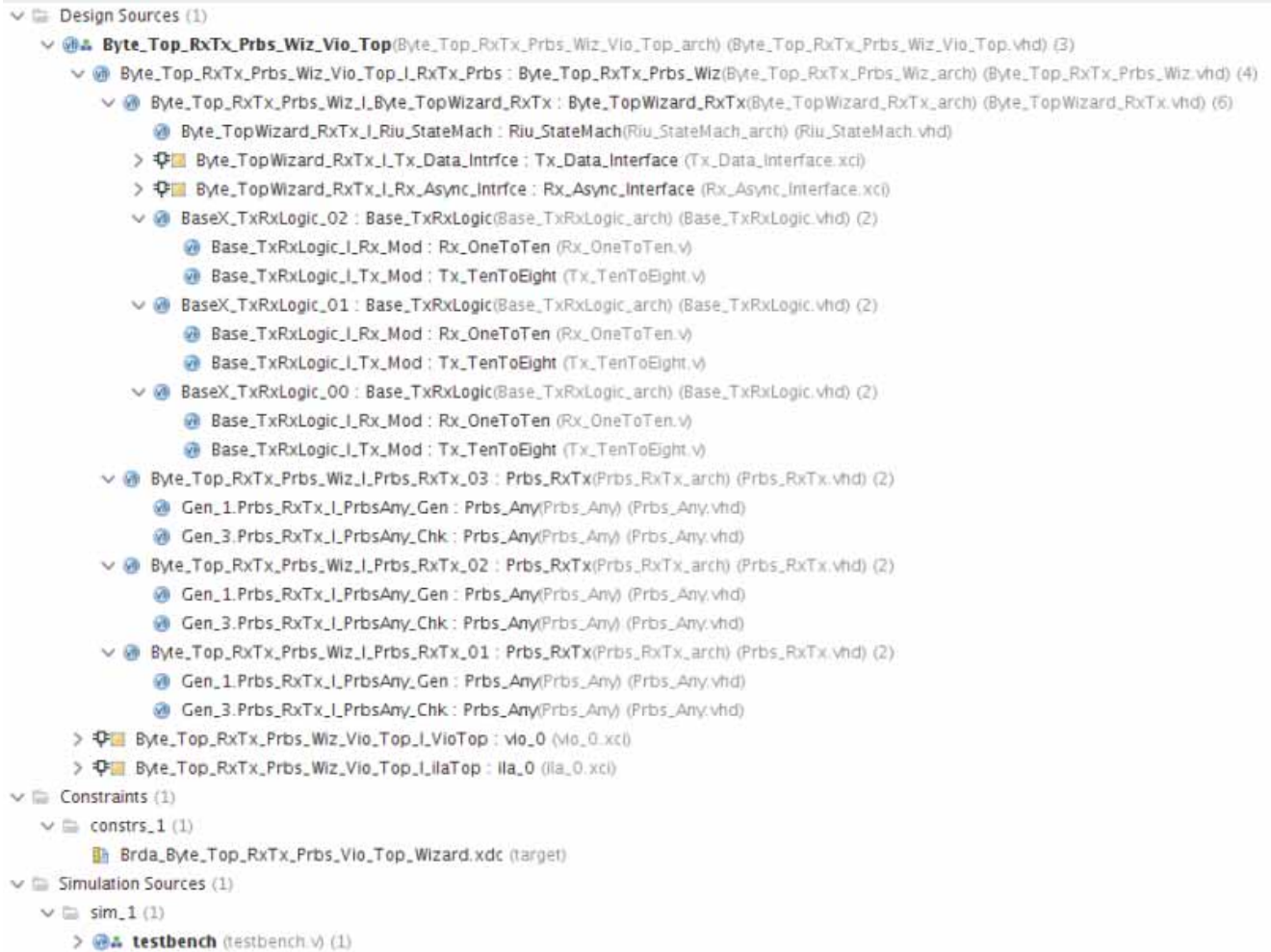
このアプリケーション ノートに付随するリファレンス デザインは、『ネイティブ モードの高速 I/O インターフェイス』(XAPP1274) のリファレンス デザインの修正版であり、High Speed SelectIO Wizard を使用して、[図 1](#) に示す TX および RX のネイティブ モード インターフェイスを生成します。



X20365-030318

図 1: 非同期のブロック図

デザイン ファイルは、[図 2](#) に示すように構成されています。ダウンロード ファイルには、プロジェクトの再構築用に TCL スクリプト (`top.tcl`) が追加されています。



X20366-030318

図 2: デザイン ファイル

リファレンス デザインは次の 4 つのコアを使用します。

- Tx\_Data\_Interface - TX\_BITSLICE のトランスミッターのセットアップ用の High Speed SelectIO Wizard
- Rx\_Async\_Interface - RX\_BITSLICE のレシーバーのセットアップ用の High Speed SelectIO Wizard
- vio\_0 - リセットを制御し、リファレンス デザインのステータスを取得するためのハードウェア インターフェイス
- ila\_0 - 動作中のデータをキャプチャするためのハードウェア デバッグ インターフェイス

制約ファイルには、ターゲット特定のデモ ボード用の追加設定があり、すべてのボードで必要でない場合もあります。High Speed SelectIO Wizard には、配置や SelectIO の設定などに対する追加制約があります。

リファレンス デザインは、『ネイティブ モードの高速 I/O インターフェイス』(XAPP1274) [参照 1] の非同期リファレンス デザインから次のように変更されています。

- RIU\_OR が移植する場合を考慮して削除されています。
- 制御信号は統合されています。
- 入力クロックは別のバンクに移動させることができます。

移植を考慮して、RIU\_OR はリファレンス デザインから削除されています。そのため、TX および RX インターフェイスは異なるバイト グループになければなりません。バイト グループで RX および TX インターフェイスを共有する必要があるデザイン用に、元の RIU\_OR は Byte\_TopWizard\_RxTx.vhd デザイン ファイル内でコメントアウトされています。RIU\_OR が削除されているので、トランスミッターとレシーバーは異なるバンクに配置できます。

デバイスのピン配置の変更は、ウィザードを使用することとポートに同じ信号名を使用することで容易になっていますが、インスタンス化テンプレートに基づいて多数のインターフェイス信号を更新する必要があります。ポート名は、特定の IP ソースのインスタンス化テンプレートで確認できます。VHDL の場合、コンポーネント定義とインスタンス化を更新する必要があります。

バイト グループに基づく RX ポート名は次のとおりです。

```
riu_addr_bg<#>      => IntRiu_Addr,           -- in [5:0]
riu_wr_data_bg<#>  => IntRiu_WrData,         -- in [15:0]
riu_rd_data_bg<#>  => IntBase_Riu_Rd_Data_Rx, -- out [15:0]
riu_valid_bg<#>    => IntBase_Riu_Valid_Rx,  -- out
riu_wr_en_bg<#>    => IntRiu_Wr_En,         -- in
riu_nibble_sel_bg<#> => IntRiu_Nibble_Sel,     -- in [1:0]
```

ニブル(ビットスライス制御)に基づく RX ポート名は次のとおりです。

```
dly_rdy_bsc<#>      => IntBase_Rx_Dly_Rdy,    -- out
```

ビットスライスに基づく RX ポート名は次のとおりです。

```
rx_en_vtc<#>        => IntBase_Rx_Bs_En_Vtc,  -- in
fifo_rd_clk<#>      => IntBase_Rx_Fifo_Rd_Clk,  -- in
fifo_rd_en<#>       => IntBase_Rx_Fifo_Rd_En_0(1), -- in
fifo_empty<#>      => IntBase_Rx_Fifo_Empty_0(1), -- out
rx_ce<#>           => Low,                    -- in
rx_inc<#>          => Low,                    -- in
rx_load<#>         => IntBase_Idly_Load_0(1),   -- in
rx_cntvaluein<#>   => IntBase_Idly_CntValueIn_0(17 downto 9), -- in [8:0]
rx_cntvalueout<#> => IntBase_Idly_CntValueOut_0(17 downto 9), -- out [8:0]
```

ニブル(ビットスライス制御)に基づく TX ポート名は次のとおりです。

```
vtc_rdy_bsc<#>     => open,                  --: OUT STD_LOGIC;
en_vtc_bsc<#>      => IntBase_Tx_Bsc_En_Vtc,  --: IN STD_LOGIC;
dly_rdy_bsc<#>     => IntBase_Tx_Dly_Rdy_bsc4, --: OUT STD_LOGIC;
```

TX ポートの場合、デザインが複数のニブルにまたがるのであれば、各ニブルの dly\_rdy\_bsc<#> は Stat\_Tx\_Dly\_Rdy (Byte\_TopWizard\_RxTx.vhd) を使用して結合する必要があります。たとえば、TX インターフェイスがニブル 0 とニブル 1 にまたがる場合、Tx\_Data\_Interface は dly\_rdy\_bsc# 接続に対して次のように調整します。

```
Stat_Tx_Dly_Rdy      <= IntBase_Tx_Dly_Rdy_bsc0 and IntBase_Tx_Dly_Rdy_bsc1;
Byte_TopWizard_RxTx_I_Tx_Data_Intrfce : Tx_Data_Interface
  port map (
    Tx_dly_rdy_bsc0 => IntBase_Tx_Dly_Rdy_bsc0,
    Tx_dly_rdy_bsc1 => IntBase_Tx_Dly_Rdy_bsc1,
    ...
  )
```

RX および TX デザインを簡略化するためのピンの選択のガイドラインを次に示します。

- 各バイト グループは RX または TX のいずれかでなければなりません (移植を考慮して RIU\_OR は簡略化)。
- 空のバイト グループは許可されます。たとえば、インターフェイスはバイト グループ 0 および 2 を使用できます。
- 各インターフェイスにまたがるバイト グループは RX または TX のいずれかでなければなりません。バイト グループ 0 および 2 を RX に使用する場合、バイト グループ 1 は RX にするかまたは空にしておく必要があります。

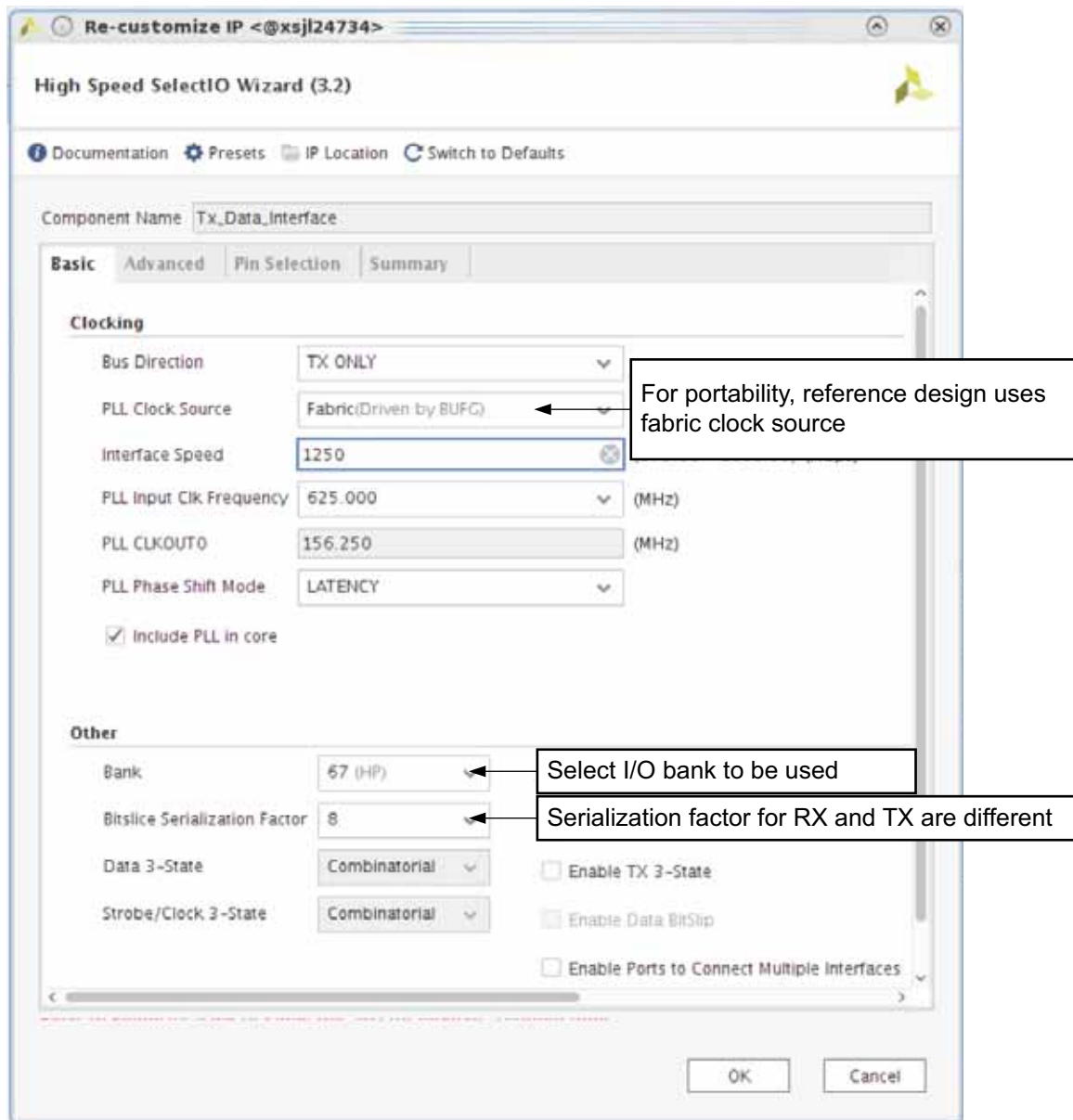
図 3 はバンク内でのビットスライスの構成を示しています。

Bitslice	Nibble (Bitslice Control)	Byte Group
51		
...		
45	7	
44		
...		
39	6	3
38		
...		
32	5	
31		
...		
26	4	2
25		
...		
19	3	
18		
...		
13	2	1
12		
...		
6	1	
5		
...		
0	0	0

X20368-030318

図 3: バンク内のビットスライス、ニブル、バイト グループ番号

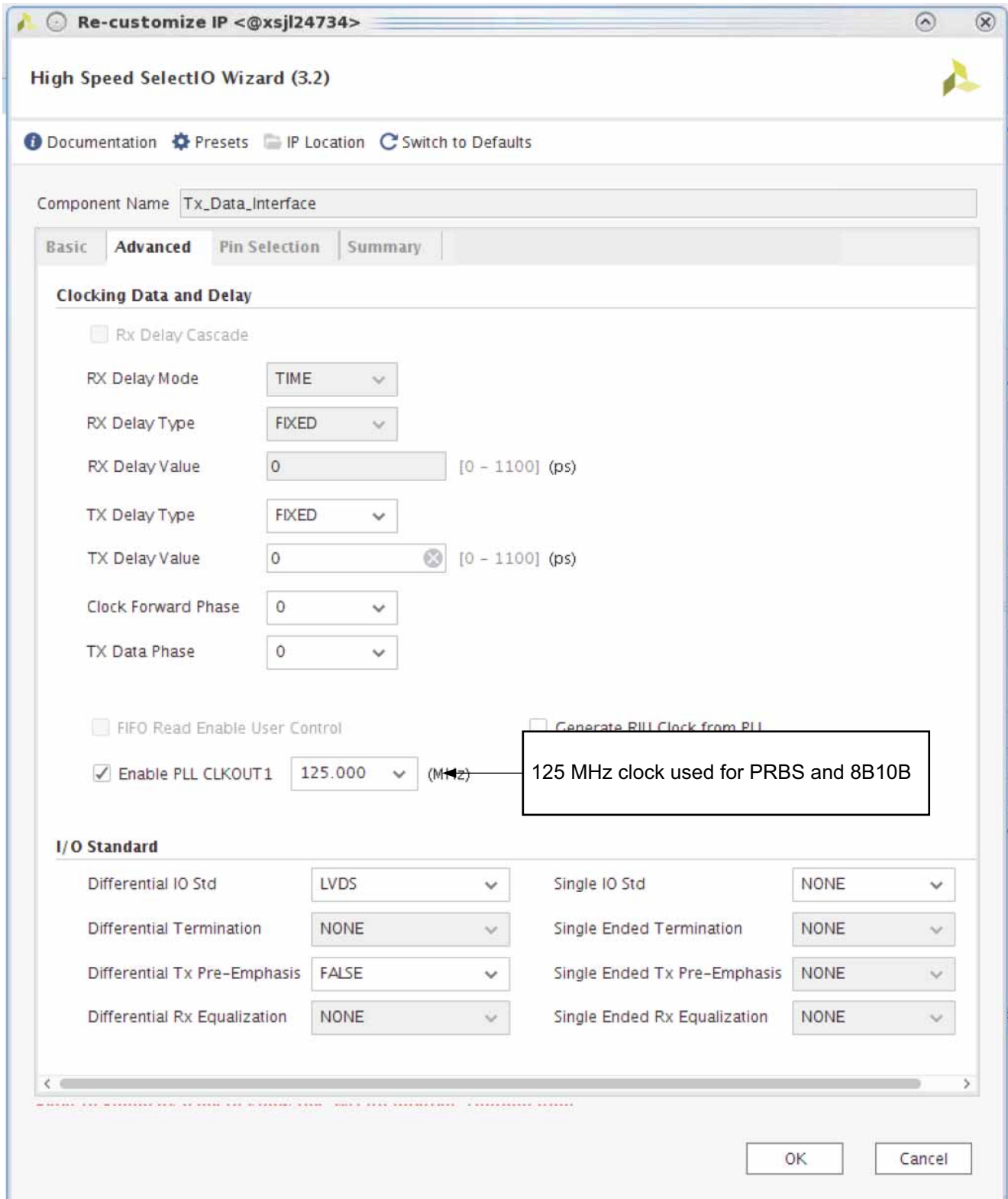
図 4 ~ 図 8 に、リファレンス デザインを機能させるための TX および RX ウィザードの設定を示します。これらの設定は、『ネイティブ モードの高速 I/O インターフェイス』(XAPP1274) [参照 1] に示されているものと同じです。



X20369-030518

図 4: High Speed SelectIO Wizard TX のセットアップ - [Basic] タブ

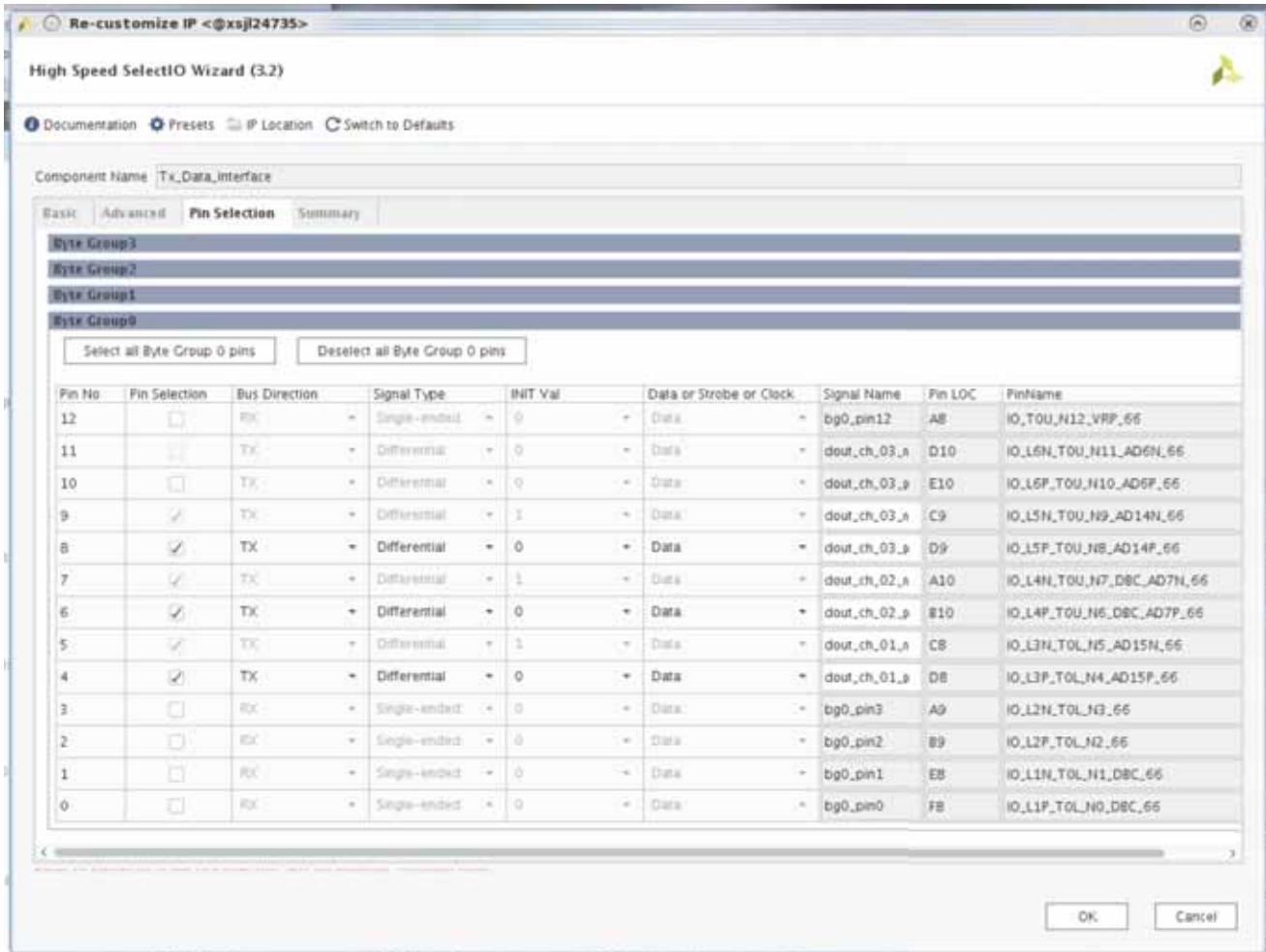




X20370-030318

図 5: High Speed SelectIO Wizard TX のセットアップ - [Advanced] タブ

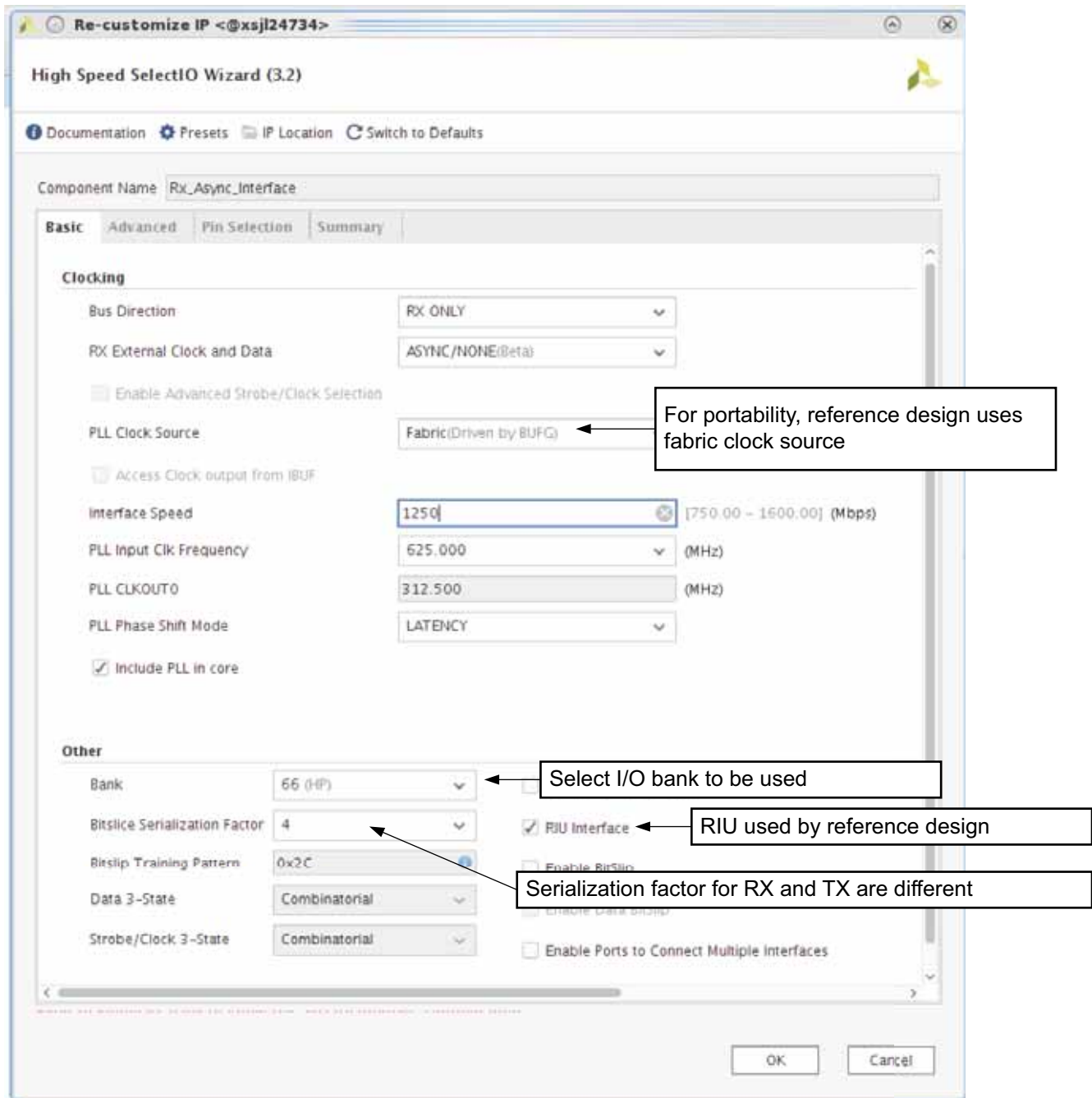




X20371-030318

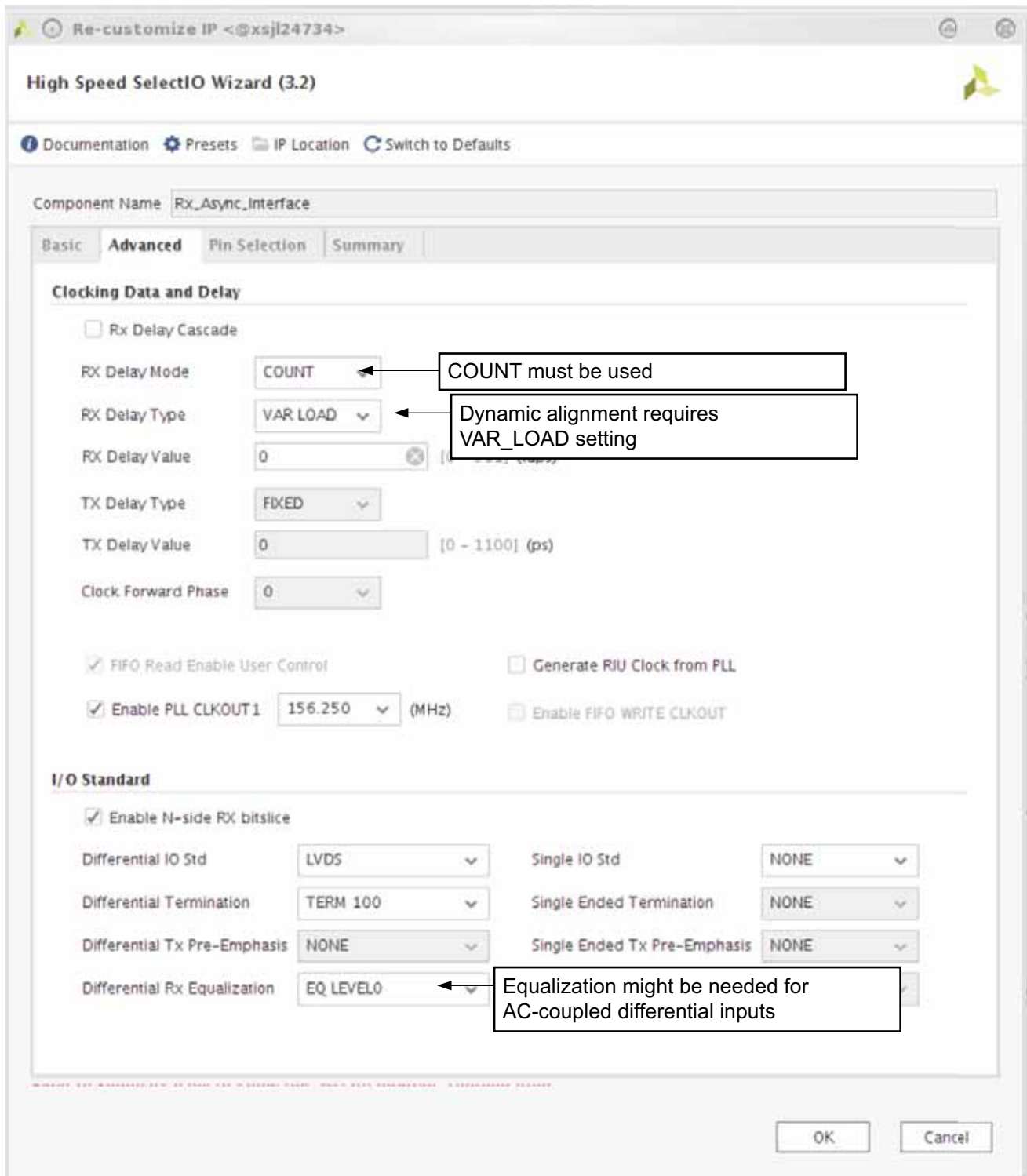
図 6: High Speed SelectIO Wizard TX のセットアップ - [Pin Selection] タブ

図 7、図 8、および図 9 に示す設定は RX で使用する必要があります。



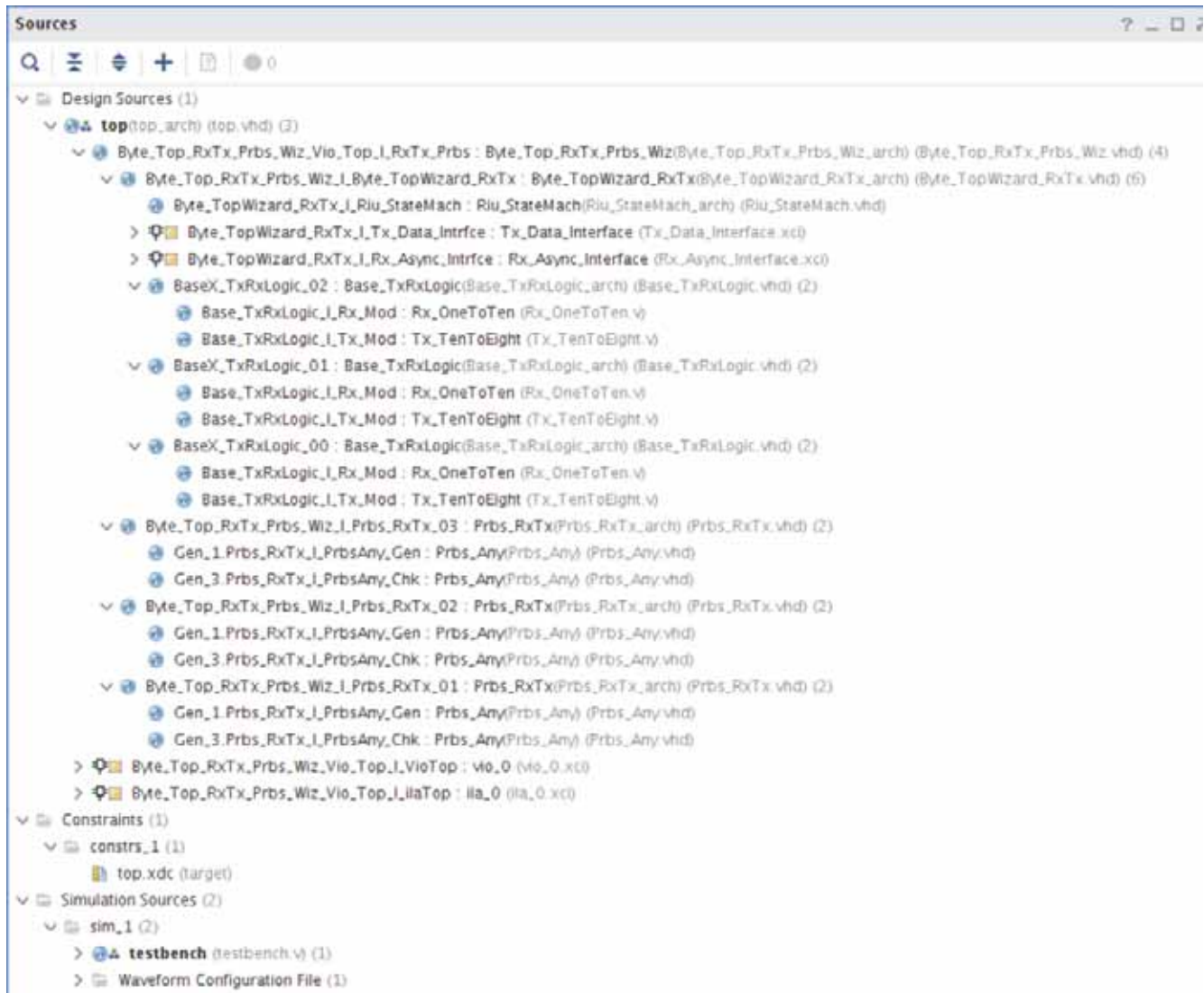
X20372-030318

図 7: High Speed SelectIO Wizard RX のセットアップ - [Basic] タブ



X20373-030518

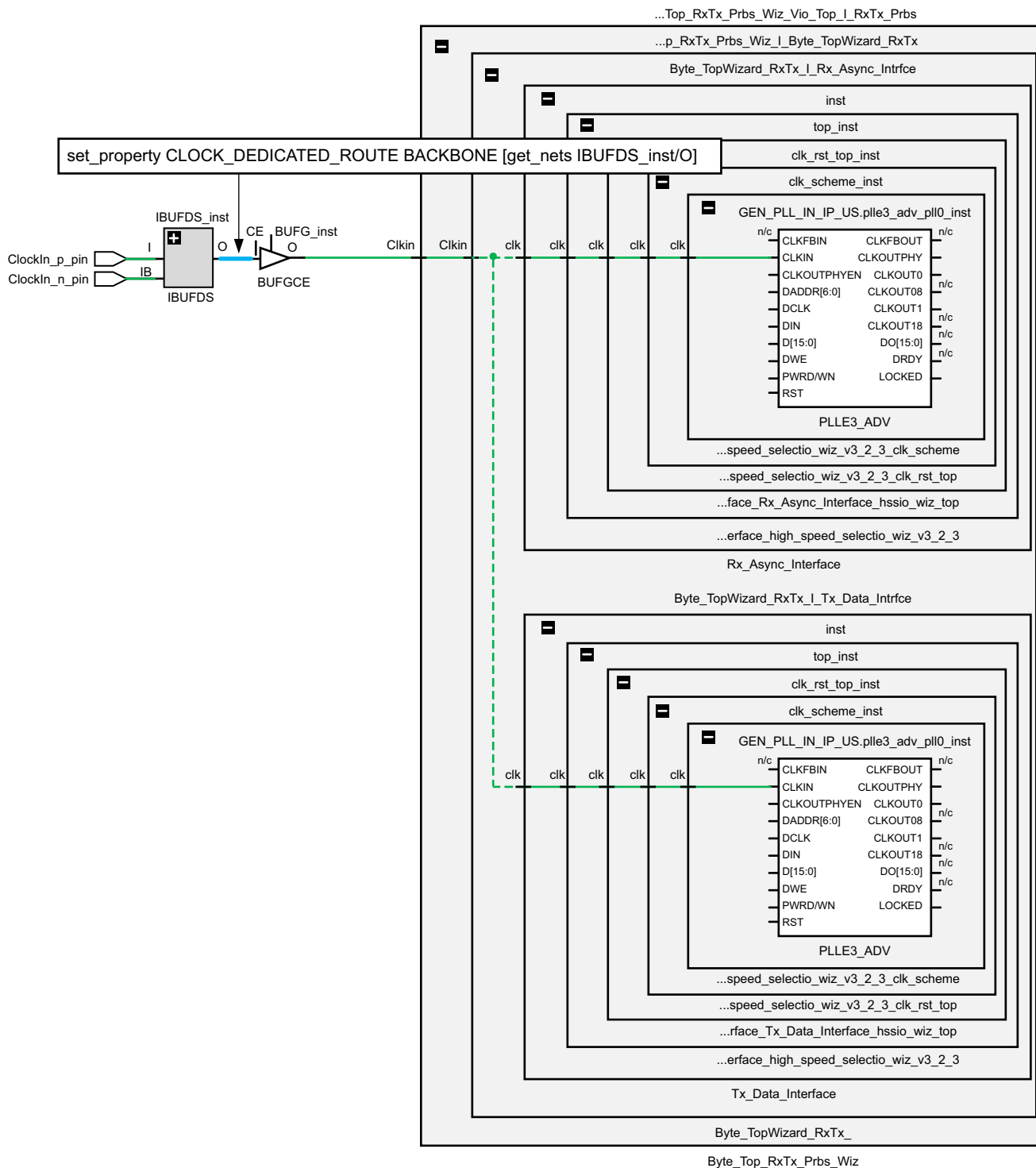
図 8: High Speed SelectIO Wizard RX のセットアップ - [Advanced] タブ



X20374-030918

図 9: High Speed SelectIO Wizard RX のセットアップ - [Pin Selection] タブ

図 10 に示すように、基準クロックを別のバンクに移動できるようにするために、追加のクロックバッファ (BUFGCE) が追加されています。クロック配線を空けるために CLOCK\_DEDICATED\_ROUTE が追加されています。この特定のケースでは、クロックはほかのロジック接続なしで PLL のみに配線されています。



X20375-030318

図 10: PLL への基準クロック接続

## シミュレーション

シンプルなテストベンチが提供されています。リファレンス デザインは、シミュレーションでサポートされていないレジスタを使用します。適切なシミュレーションのために、最上位デザイン ファイル `Byte_Top_RxTx_Prbs_Wiz_Vio_Top.vhd` で `C_InSimulation = "true"` を設定します。

```
Byte_Top_RxTx_Prbs_Wiz_Vio_Top_I_RxTx_Prbs : entity
xil_defaultlib.Byte_Top_RxTx_Prbs_Wiz
  generic map (
    C_InSimulation          => "true",
    C_SimDevice             => "ULTRASCAL"
  )
```

リファレンス デザインは、シミュレーションでサポートされていない予約済み `BITSLICE_CONTROL` レジスタ インターフェイス ユニット (RIU) にアクセスします。 `Riu_StateMach.vhd` で `C_InSimulation = "true"` の場合、 `Rx_BtVal` は 160 (0xA0) に設定されます。 `C_InSimulation = "true"` は、合成後およびインプリメンテーション後のシミュレーションで `Rx_BtVal` を固定します。

シミュレーションが完了したら、 `C_InSimulation = "false"` を設定して、 `Rx_BtVal` が `BITSLICE_CONTROL` から読み出されるようにします。 `BITSLICE_CONTROL` に接続されている `Rx_RdData` から `Rx_BtVal` が読み出されるように、デザインは後で再合成する必要があります。

RIU ステート マシンをシンプルにして移植性を高めるため、 `Rx_BtVal` のビット タイムをすべてのレシーバーのビット タイムとして使用します (各レシーバーは同じデータ レートで実行しているため)。 `Byte_TopWizard_RxTx.vhd` では、 `Int_Base_Rx_BtVal` が各 `BaseX_TxRxLogic` のインスタンスに使用されます。

## ハードウェア動作

リファレンス デザインは、FMC HPC コネクタを使用して KCU105 ボード用にセットアップされています。XM107 FMC ループバック カードを J22 FMC コネクタに接続します。XM107 は LA[16:0] を LA[33:17] に接続します。『FMC XM107 ループバック カード ユーザー ガイド』 (UG539) [参照 3] を参照してください。

デザインには 2 つのクロック ソースが必要です。VIO インターフェイスは 100MHz クロックに直接接続され、VIO インターフェイスを制御したり、デザインを制御するための入力/出力信号を生成したりします。

トランスミッターとレシーバーのチャネルは、 `SI570_CLK_SEL = 1` のときに、 `Si570` に接続される 625MHz クロックが必要です。クロックを制御するには、システム コントローラーを使用して、シリアル ポートを『KCU105 ボード ユーザー ガイド』 (UG197) [参照 2] の説明に従って使用します。シリアル ポートを使用することで、システム コントローラーは KCU105 `Si570` ユーザー クロックの周波数を 625MHz に設定できます。

一部のシステムでは、シリアル ポートを使用して FMC へも電源供給が必要になる場合があります。『KCU105 ボード ユーザー ガイド』 (UG917) などのボード ユーザー ガイドの手順に従ってください。ハードウェア動作は、図 11 に示すとおり、VIO 出力を使用したりリセット シーケンスに従う必要があります。

Name	Value	Active
SI570_CLK_SEL_OBUF[0:0]	[B] 0	
EnaPrbsGenerator_01	[B] 1	
EnaPrbsGenerator_02	[B] 1	
EnaPrbsGenerator_03	[B] 1	
IntResetIn	[B] 0	
Rst_RxTxLogic_vio	[B] 0	
PrbsRst_Chnl_01	[B] 0	
PrbsRst_Chnl_02	[B] 0	
PrbsRst_Chnl_03	[B] 0	
PrbsClearSticky_01	[B] 0	
PrbsClearSticky_02	[B] 0	
PrbsClearSticky_03	[B] 0	
InjectError_Chnl_01	[B] 0	
InjectError_Chnl_02	[B] 0	
InjectError_Chnl_03	[B] 0	

Callout boxes:

- KCU105 multiplexer select SI570 (0) or SI5328 (1)
- Transmitter enable to send PRBS data
- IntResetIn - master reset for PLLs, transmitter, pattern generator, receiver, and pattern checker
- Rst\_TxLogic used to reset bitslice logic
- Resets error flags and PrbsErrCnt\_<03|02|01>
- Clears error flag for PrbsClearSkicky\_<03|02|01>
- Inject transmitter errors into the PRBS data. Used to ensure PRBS checker and receive interface are correctly capturing errors in transmission.

X20376-030518

図 11: 仮想 I/O (VIO) 出力

リセットシーケンスは簡易化されており、IntResetIn が High にアサートされるとリセットされます。『ネイティブモードの高速 I/O インターフェイス』(XAPP1274) [参照 1] との互換性のために、図 11 に示すとおり、リセットとイネーブルの制御が引き続き使用可能です。

正しく動作すると、GernatePrbsValid\_<03|02|01> と PrbsErrorDetec<03|02|01> のトグルが継続し、インターフェイスが稼働していることを示します。



The screenshot shows a window titled "dashboard\_inputs" with a sub-window "hw\_vio\_1". It displays a table of hardware I/O signals. Annotations provide context for specific signals:

- CR\_Debug\_Out\_1[43:0]**: Value is [H] 000\_0001\_6EAE. An annotation points to this value, stating "[17:9] Bt\_Val" and "[8:0] State machine".
- GeneratPrbsValid\_01, \_02, \_03**: All have a value of [B] 0. An annotation for GeneratPrbsValid\_01 states: "GeneratPrbsValid with activity indicates transmitter is sending PRBS data".
- PrbsErrCnt\_01[63:0], \_02[63:0], \_03[63:0]**: All have a value of [H] 0000\_0005\_6E27\_EBC1. An annotation for PrbsErrCnt\_01 states: "PrbsErrCnt should be running for normal operation".
- PrbsErrDetSticky\_01, \_02, \_03**: All have a value of [B] 0. An annotation for PrbsErrDetSticky\_01 states: "PrbsErrDetSticky and PrbsErrDetect are Low with no activity".

Name	Value	Acti...	Direc...	VIO
> CR_Debug_Out_1[43:0]	[H] 000_0001_6EAE		Input	hw_vio_1
↳ GeneratPrbsValid_01	[B] 0	↕	Input	hw_vio_1
↳ GeneratPrbsValid_02	[B] 0	↕	Input	hw_vio_1
↳ GeneratPrbsValid_03	[B] 0	↕	Input	hw_vio_1
> PrbsErrCnt_01[63:0]	[H] 0000_0005_6E27_EBC1	↕	Input	hw_vio_1
> PrbsErrCnt_02[63:0]	[H] 0000_0005_6E27_EBC1	↕	Input	hw_vio_1
> PrbsErrCnt_03[63:0]	[H] 0000_0005_6E27_EBC1	↕	Input	hw_vio_1
↳ PrbsErrDetSticky_01	[B] 0		Input	hw_vio_1
↳ PrbsErrDetSticky_02	[B] 0		Input	hw_vio_1
↳ PrbsErrDetSticky_03	[B] 0		Input	hw_vio_1
↳ PrbsErrorDetect_01	[B] 0		Input	hw_vio_1
↳ PrbsErrorDetect_02	[B] 0		Input	hw_vio_1
↳ PrbsErrorDetect_03	[B] 0		Input	hw_vio_1
↳ Rx_LogicRst_1	[B] 0		Input	hw_vio_1
↳ Tx_LogicRst	[B] 0		Input	hw_vio_1

X20377-030318

図 12: 仮想 I/O (VIO) 入出力

PrbsErrDetSticky\_<03|02|01> および PrbsErrCnt\_<03|02|01> にエラーがないかを調べます。

インターフェイスが動作していることを確認するために、InjectError\_Chnl\_<03|02|01> を使用して特定のチャンネルでエラーを挿入します。図 13 に示すとおり、エラーが発生すると PrbsErrDetSticky は High のままになります。この例では、InjectError\_Chnl\_02 が High にアサートされています。エラーが発生すると、PrbsErrorDetect\_02 は一瞬 High になり、動作も値の変更を示します。一定量のエラーが発生すると、PrbsErrCnt は動作を停止して終了します。

Name	Value	Acti...	Directi...	VIO
CR_Debug_Out_1[43:0]	[H] 000_0001_6EAE		Input	hw_vio_1
GeneratPrbsValid_01	[B] 0	↓	Input	hw_vio_1
GeneratPrbsValid_02	[B] 0	↓	Input	hw_vio_1
GeneratPrbsValid_03	[B] 0	↓	Input	hw_vio_1
PrbsErrCnt_01[63:0]	[H] 0000_000F_E680_CBE8	↓	Input	hw_vio_1
PrbsErrCnt_02[63:0]	[H] 0000_000B_15E3_A44D	↓	Input	hw_vio_1
PrbsErrCnt_03[63:0]	[H] 0000_000F_E680_CBE8	↓	Input	hw_vio_1
PrbsErrDetSticky_01	[B] 0		Input	hw_vio_1
PrbsErrDetSticky_02	[B] 1		Input	hw_vio_1
PrbsErrDetSticky_03	[B] 0		Input	hw_vio_1
PrbsErrorDetect_01	[B] 0		Input	hw_vio_1
PrbsErrorDetect_02	[B] 1	↑	Input	hw_vio_1
PrbsErrorDetect_03	[B] 0		Input	hw_vio_1
Rx_LogicRst_1	[B] 0		Input	hw_vio_1
Tx_LogicRst	[B] 0		Input	hw_vio_1

X20378-030318

図 13: 挿入されたエラー後の仮想 I/O (VIO) 入力

エラーが発生した後は、リセットシーケンスに従います。PrbsClearSticky\_<03|02|01> および PrbsRst\_Chnl\_<03|02|01> は、エラーフラグとエラーカウンタをクリアします。

注記: PrbsErrCnt\_<03|02|01> にアクティビティがなく、[H]0000\_0000\_0000\_0000 である場合には、FMC の電源供給が設定されていることを確認します。FMC の電源供給の設定方法については、特定のボードのユーザーガイドを調べてください。

## まとめ

非同期リファレンス デザインは、RX\_BITSLICE および BITSLICE\_CONTROL プリミティブを制御するネイティブプリミティブで機能します。これらのアルゴリズムは、『ネイティブモードの高速 I/O インターフェイス』(XAPP1274) [参照 1] で説明されています。結果として、ピンが High Speed SelectIO Wizard で変更されている場合、ポート接続を更新する必要があります。このアプリケーション ノートのガイドラインに従って、リファレンス デザインのピン配置を変更できます。また、デザインの使用に関する追加説明も記載されています。

# Xilinx Documentation Navigator およびデザイン ハブ

Xilinx Documentation Navigator (DocNav) では、ザイリンクスの資料、ビデオ、サポート リソースにアクセスでき、特定の情報を取得するためにフィルター機能や検索機能を利用できます。DocNav を開くには、次のいずれかを実行します。

- Vivado IDE で [Help] [Documentation and Tutorials] をクリックします。
- Windows で [スタート] [すべてのプログラム] [Xilinx Design Tools] [DocNav] をクリックします。
- Linux コマンド プロンプトに「docnav」と入力します。

ザイリンクス デザイン ハブには、資料やビデオへのリンクがデザイン タスクおよびトピックごとにまとめられており、これらを参照することでキー コンセプトを学び、よくある質問 (FAQ) を参考に問題を解決できます。デザイン ハブにアクセスするには、次のいずれかを実行します。

- DocNav で [Design Hubs View] タブをクリックします。
- ザイリンクス ウェブサイトの [デザイン ハブ](#) ページを参照します。

注記: DocNav の詳細は、ザイリンクス ウェブサイトの [Documentation Navigator](#) ページを参照してください。



注意: DocNav から、日本語版は参照できません。ウェブサイトのデザイン ハブ ページをご利用ください。

## 参考資料

注記: 日本語版のバージョンは、英語版より古い場合があります。

1. 『ネイティブ モードの高速 I/O インターフェイス』 (XAPP1274: [英語版](#)、[日本語版](#))
2. 『KCU105 ボード ユーザー ガイド』 ([UG917](#))
3. 『FMC XM107 ループバック カード ユーザー ガイド』 ([UG539](#))
4. 『LogiCORE IP High Speed SelectIO Wizard 製品ガイド』 (PG188: [英語版](#)、[日本語版](#))
5. 『UltraScale アーキテクチャ SelectIO リソース』 (UG571: [英語版](#)、[日本語版](#))

## 改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂内容
2018 年 3 月 29 日	1.0	初版

## 重要な法的通知

本通知に基づいて貴殿または貴社(本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」、以下同じ)に開示される情報(以下「本情報」といいます)は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1)本情報は「現状有姿」、およびすべて受領者の責任で(with all faults)という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず(商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない(否認する)ものとします。また、(2)ザイリンクスは、本情報(貴殿または貴社による本情報の使用を含む)に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない(契約上、不法行為上(過失の場合を含む)、その他のいかなる責任の法理によるかを問わない)ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害(第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます)が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、<https://japan.xilinx.com/legal.htm#tos>で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うこととなります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。<https://japan.xilinx.com/legal.htm#tos>で見られるザイリンクスの販売条件を参照してください。

### 自動車用のアプリケーションの免責条項

オートモーティブ製品(製品番号に「XA」が含まれる)は、ISO 26262 自動車用機能安全規格に従った安全コンセプトまたは余剰性の機能(「セーフティ設計」)がない限り、エアバッグの展開における使用または車両の制御に影響するアプリケーション(「セーフティアプリケーション」)における使用は保証されていません。顧客は、製品を組み込むすべてのシステムについて、その使用前または提供前に安全を目的として十分なテストを行うものとします。セーフティ設計なしにセーフティアプリケーションで製品を使用するリスクはすべて顧客が負い、製品の責任の制限を規定する適用法令および規則にのみ従うものとします。

© Copyright 2018 Xilinx, Inc. Xilinx, Xilinx のロゴ、Artix、ISE、Kintex、Spartan、Virtex、Vivado、Zynq、およびこの文書に含まれるその他の指定されたブランドは、米国およびその他の各国のザイリンクス社の商標です。すべてのその他の商標は、それぞれの所有者に帰属します。この資料に関するフィードバックおよびリンクなどの問題につきましては、[jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com)まで、または各ページの右下にある[フィードバック送信]ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメールアドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。[mailto:jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com)

この資料に関するフィードバックおよびリンクなどの問題につきましては、[jpn\\_trans\\_feedback@xilinx.com](mailto:jpn_trans_feedback@xilinx.com)まで、または各ページの右下にある[フィードバック送信]ボタンをクリックすると表示されるフォームからお知らせください。フィードバックは日本語で入力可能です。いただきましたご意見を参考に早急に対応させていただきます。なお、このメールアドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。