



XAPP1275 (v1.0) 2016 年 1 月 27 日

Kintex UltraScale FPGA GTH トランシーバーを使用した HDMI 2.0 の実装

著者 : Gilbert Magnaye, Marco Groeneveld

概要

このアプリケーション ノートでは、Kintex® UltraScale™ FPGA GTH トランシーバーと次のザイリンクス LogiCORE™ IP の機能を使用して HDMI™ (High-Definition Multimedia Interface) 2.0 を実装する際の考慮事項について説明します。

- HDMI 1.4/2.0 Transmitter Subsystem
- HDMI 1.4/2.0 Receiver Subsystem
- Video PHY Controller

デザインは、HDMI ソリューション用に送信専用モードとパススルー モードの 2 つの動作モードを備えています。送信専用モードでは、LogiCORE IP テスト パターン ジェネレーター (TPG) コアからのカラー バー パターンを表示します。パススルー モードでは、外部 HDMI ソースを使用してビデオ データを HDMI レシーバー経由で送信し、その後 HDMI トランスミッターから外部 HDMI シンクに送信して表示します。

リファレンス デザインは、Kintex UltraScale XCKU040-2FFVA1156 FPGA と inrevium TB-FMCH-HDMI4K [参照 1] ドーターカードを使用するザイリンクス Kintex UltraScale FPGA KCU105 評価ボード [参照 2] をターゲットとしています。

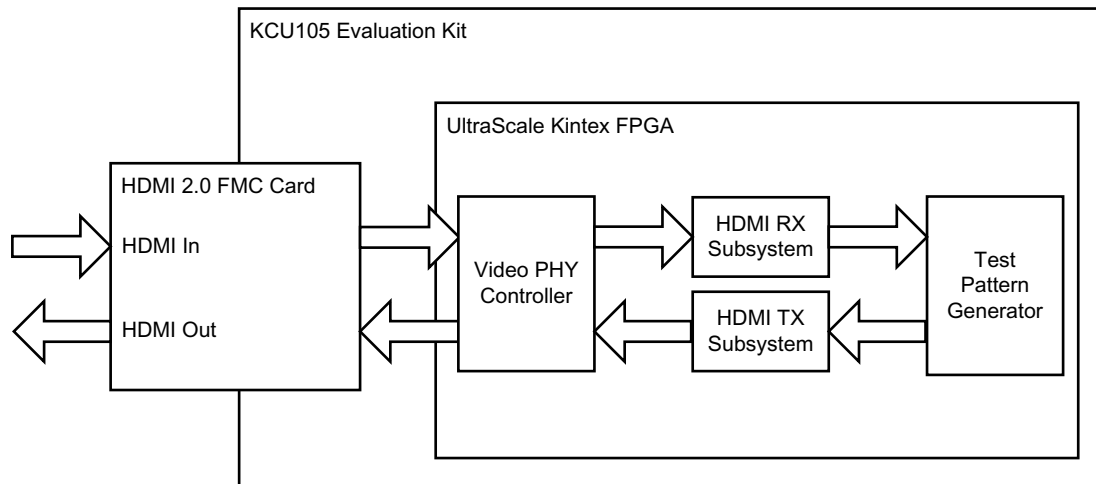
このアプリケーション ノートの [リファレンス デザイン ファイル](#) は、ザイリンクスのウェブサイトからダウンロードできます。デザイン ファイルの詳細は、「[リファレンス デザイン](#)」を参照してください。

リファレンス デザイン

リファレンス デザインは、Vivado® Design Suite System Edition 2015.4 を使用して作成および構築されています。デザインには、ザイリンクスのソフトウェア開発キット (SDK) 2015.4 を使用して構築されたソフトウェアも含まれます。このソフトウェアは、MicroBlaze™ プロセッサ サブシステムで実行され、制御機能とステータス機能を実装します。このアプリケーション ノートでは Vivado Design Suite および SDK の完全なプロジェクト ファイルを提供しており、デザインの検討や再構築に活用したり、新規デザインのテンプレートとして使用できます。

ハードウェア

リファレンス デザインは、HDMI 1.4/2.0 Transmitter Subsystem (HDMI_TX_SS)、HDMI 1.4/2.0 Receiver Subsystem (HDMI_RX_SS)、および Video PHY (VPHY) Controller コアを中心に、既存のザイリンクス IP コアを組み合わせることで完全なシステムを構成しています。システムの入出力は、ザイリンクス開発ボードの FMC HPC コネクタに接続される HDMI 2.0 ドーターカードを介した HDMI ビデオストリームとなります。図 1 を参照してください。



X15808-012216

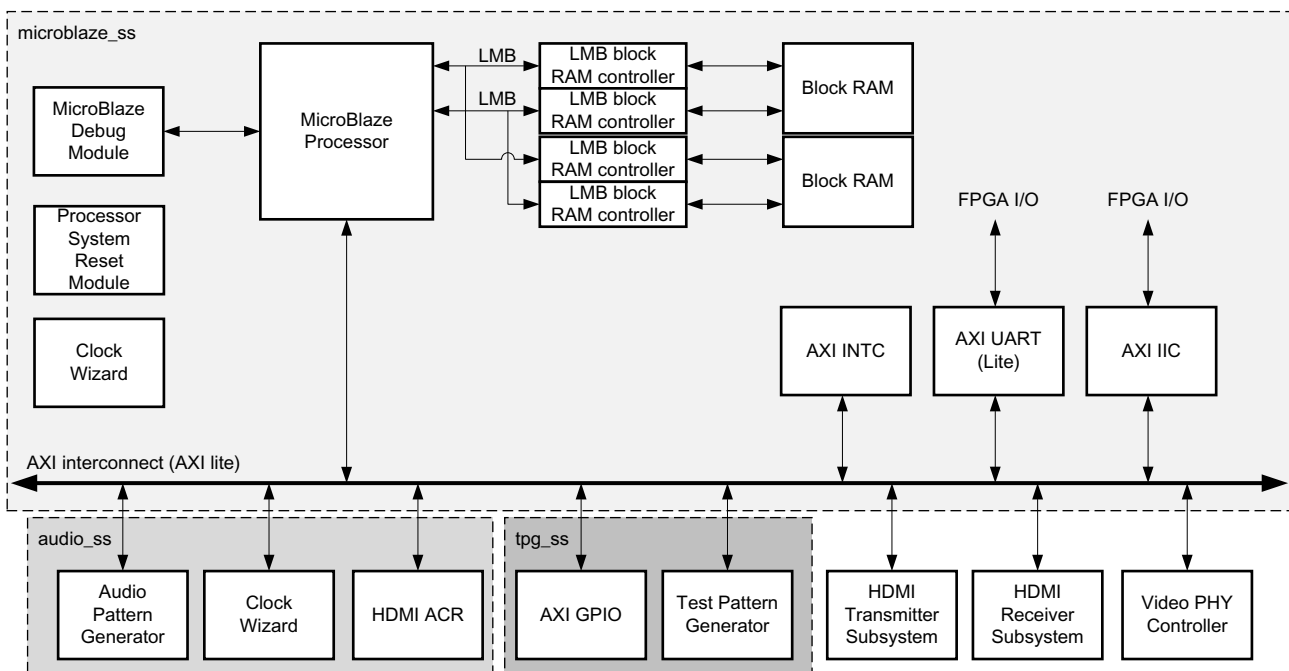
図 1: KCU105 ボードを使用した HDMI リファレンス デザインのブロック図

VPHY Controller コアは、HDMI 2.0 ドーターカードとの間でネイティブ HDMI ビデオの送受信を可能にする HDMI アプリケーション用にコンフィギュレーションされています。

パススルーモードでは、VPHY Controller コアは高速シリアルビデオストリームを復元して、パラレルデータストリームに変換し、HDMI_RX_SS コアに転送します。HDMI_RX_SS コアは、HDMI ストリームからビデオストリームとオーディオストリームを抽出し、別々の AXI ビデオストリームとオーディオストリームに変換します。AXI ビデオストリームは TPG コアを通り、AXI オーディオストリームはカスタマイズされたオーディオ生成ブロックを通ります。2つの AXI ストリームは、最終的に HDMI_TX_SS コアに到達します。HDMI_TX_SS コアは、AXI ビデオストリームとオーディオストリームを再び HDMI ストリームに変換します。その後、VPHY Controller コアが、このストリームを高速シリアルデータストリームとして送信します。HDMI 入力インターフェイスからの TMDS (Transition Minimized Differential Signaling) クロックは、HDMI 2.0 FMC カード内の SI5323 クロックジェネレーターを介して HDMI TX トランシーバーに転送されます。

送信専用モードでは、TPG コアがカラーバーパターンを AXI ビデオストリームとして生成し、カスタマイズされたオーディオ処理ブロックが低周波数オーディオを AXI オーディオストリームとして生成します。2つのストリームは、HDMI_TX_SS コアに転送されて HDMI ストリームに変換された後、VPHY Controller コアに転送されて送信されます。これに対応する HDMI TX トランシーバーの基準クロックは、HDMI 2.0 FMC カード内の SI5323 クロックジェネレーターによって生成されます。

I/O ペリフェラルおよびプロセッサ サポート IP を含むシステム レベルの制御は、簡略化した MicroBlaze エンベデッド プロセッサ サブシステムが担います。クロック ジェネレーター ブロックとプロセッサ システム リセット ブロックは、システム全体にクロック信号とリセット信号をそれぞれ供給します。図 2 および表 1 に、MicroBlaze プロセッサ サブシステムのブロック図とアドレス マップを示します。



X15809-012216

図 2 : MicroBlaze プロセッサ サブシステムのブロック図

表 1 : MicroBlaze サブシステムのアドレス マップ

ペリフェラル	インスタンス	ベース アドレス	上位アドレス
lmb_bram_if_cntlr	ilmb_bram_if_cntlr_0	0x00000000	0x0007FFFF
lmb_bram_if_cntlr	dlmb_bram_if_cntlr_0	0x00000000	0x0007FFFF
lmb_bram_if_cntlr	ilmb_bram_if_cntlr_1	0x00080000	0x0009FFFF
lmb_bram_if_cntlr	dlmb_bram_if_cntlr_1	0x00080000	0x0009FFFF
axi_intc	axi_intc_0	0x41200000	0x4120FFFF
axi_uartlite	axi_uartlite_0	0x40600000	0x4060FFFF
axi_iic	fmch_axi_iic_0	0x40800000	0x4080FFFF
aud_pat_gen	aud_pat_gen_0	0x44A60000	0x44A6FFFF
clk_wiz	clk_wiz_0	0x44A50000	0x44A5FFFF
hdmi_acr_ctrl	hdmi_acr_ctrl_0	0x44A70000	0x44A7FFFF
axi_gpio	axi_gpio_0	0x44A90000	0x44A9FFFF
v_tpg	v_tpg_0	0x44A00000	0x44A0FFFF
v_hdmi_rx_ss	v_hdmi_rx_ss_0	0x44B00000	0x44B3FFFF
v_hdmi_tx_ss	v_hdmi_tx_ss_0	0x44C00000	0x44C3FFFF
vid_phy_controller	vid_phy_controller_0	0x44A10000	0x44A1FFFF

その他のコア

VPHY、HDMI_TX_SS、HDMI_RX_SS の各コアのほかに、リファレンス デザインには次のコアが含まれています。

- MicroBlaze
- MicroBlaze Debug Module
- AXI Interconnect
- Local Memory Bus
- LMB Block RAM Controller
- Block Memory Generator
- Clocking Wizard
- Processor System Reset
- AXI UARTLITE
- AXI Interrupt Controller
- AXI IIC
- AXI GPIO
- Test Pattern Generator
- Concat
- AXI4-Stream Register Slice
- Utility Buffer
- Utility Vector Logic
- AUD_PAT_GEN (カスタム IP)
- HDMI_ACR_CTRL (カスタム IP)
- ODDR (カスタム IP)
- HDMI_HB (カスタム IP)

ハードウェア システム

このセクションでは、主要 IP ブロックの使用方法など、リファレンス デザインの機能の概要を説明します。

Video PHY Controller コア

VPHY Controller コアは、標準化されたインターフェイスを介して HDMI TX および RX (Video MAC) Subsystem IP コアとのプラグアンドプレイ接続を実現します。AXI4-Lite レジスタ インターフェイスにより、クロック検出器および NI-DRU (Non-Integer Data Recovery Unit) などの VPHY Controller コア ペリフェラルとトランシーバーに動的にアクセスできます。

高速シリアル データの物理的送受信のほか、VPHY Controller コアは次の機能を備えています。

- HDMI_TX_SS コアと HDMI_RX_SS コアに必要なクロック (リンク クロックおよびビデオ クロック) を生成します。各クロックは、『HDMI 1.4/2/0 Transmitter Subsystem LogiCORE IP 製品ガイド』(PG235) [参照 3] および『HDMI 1.4/2/0 Receiver Subsystem LogiCORE IP 製品ガイド』(PG236) [参照 4] の「クロッキング」の記述に従って生成されます。
- 差動 HDMI TX インターフェイス クロックを生成します。
- パススルー モードでのクロック転送のために、差動 HDMI RX TMDS インターフェイス クロックを再送信します。
- HDMI_TX_SS (vid_phy_status_sb_tx インターフェイス) および HDMI_RX_SS (vid_phy_status_sb_rx インターフェイス) にステータス側帯波信号を提供し、内部ステート マシンを実現します。側帯波信号は次の 2 つから成ります。
 - vid_phy_status_sb_rx_tdata[0] - リンク クロック ロック ステータス信号
 - vid_phy_status_sb_rx_tdata[1] - ビデオ クロック ロック ステータス信号
- クロック検出器ペリフェラルを介して TX および RX TMDS クロックの実際の周波数を検出します。

- NI-DRU を使用して、TMDS クロック周波数が受信位相ロック ループ (PLL) でサポートされる最小基準クロックよりも低い、HDMI の低ライン レートを復元します。

VPHY Controller コアの詳細は、『Video PHY Controller LogiCORE IP 製品ガイド』(PG230) [参照 5] を参照してください。

UltraScale GTH VPHY Controller コアの実装

ザイリンクス UltraScale FPGA の GTH トランシーバーは、QPLL0、QPLL1、CPLL の 3 種類の PLL を備えています。QPLL0 と QPLL1 は、トランシーバー クワッド内の 4 つのトランシーバーすべてに共有されます。CPLL は各トランシーバーにあります。VPHY Controller コアは、すべての種類の PLL を使用してトランスミッターとレシーバーの動作を同時にサポートします。VPHY Controller コアによって、トランスミッターが QPLL0/1 または CPLL のどちらかを使用するかを設計者が選択できます。レシーバーは、トランスミッターが使用しない PLL を自動的に使用します。

HDMI レシーバーに CPLL を使用する場合は、一定の制限が課せられます。GT ドライバーはオーバーサンプリング手法を使用して CPLL の制限を回避するため、TX にはこれらの制限はありません。ここでは HDMI レシーバーに CPLL を使用する場合の制限について説明します。

CPLL の電圧制御オシレーター (VCO) は、2.0GHz ~ 6.25GHz の範囲で動作しなければなりません。VCO の周波数は、TMDS クロック周波数によって決まります。CPLL は、限られた一連の通倍値を TMDS クロック周波数に適用できます。GT ドライバーは、TMDS クロック周波数を測定し、VCO の周波数が許容範囲内に収まるような有効な通倍値を見つけます。CPLL が適用できる最大通倍値は 20 なので、CPLL がサポートする最小 TMDS クロック周波数は 100MHz になります。CPLL は、TMDS クロック周波数が 100MHz より低いビデオ フォーマットをサポートしていません。GT ドライバーは、TMDS クロック周波数が 100MHz より低いことを検出すると、NI-DRU を有効にして 1Gb/s 未満の低ビット レートを受信します。NI-DRU は 2.5Gb/s で動作するため、CPLL がサポートしていないライン レートを復元できます。TMDS クロック周波数が 100MHz より高い場合は、表 2 に示すように 10 または 20 の通倍値を適用して、VCO の周波数を適切な範囲内に保ちます。

表 2 : UltraScale GTH の CPLL の使用方法

TMDS クロック周波数 (MHz)	CPLL 基準クロックの分周値	CPLL の通倍値	VCO の周波数	注記
<100	N/A	N/A	N/A	TX : オーバーサンプリング RX : NI-DRU が使用される
100 ~ 312.5	1	20	2.0 ~ 6.25GHz	CDR が使用される
312.5 ~ 340	1	10	3.125 ~ 3.4GHz	CDR が使用される



重要 : CPLL を使用する場合、HDMI RX は、NI-DRU またはネイティブ CDR を使用して、最大ライン レート 6Gb/s までの有効なビデオ フォーマットのほとんどを受信できます。

CPLL には、10BPC の 2160p24/25/30 フォーマットの TMDS 基準クロックは 100MHz より低速で、ライン レートは 2.5Gb/s より高いため、これらのフォーマットの送受信に使用できないという制限があります。また、設定可能な CPLL 分周値に制限があるため、オーバーサンプリング手法は使用できません。

表 3 : CPLL による RGB および YCbCr 4:4:4 ビデオ フォーマットのサポート

解像度	ピクセルあたりのビット数			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	DRU	DRU	√
1080i50	DRU	DRU	√	√
1080i60	DRU	DRU	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	DRU	DRU	√	√
720p60	DRU	DRU	√	√

表 3 : CPLL による RGB および YCbCr 4:4:4 ビデオ フォーマットの サポート (続き)

解像度	ピクセルあたりのビット数			
	24	30	36	48
1080p24	DRU	DRU	√	√
1080p25	DRU	DRU	√	√
1080p30	DRU	DRU	√	√
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	(1)	√	√
2160p25	√	(1)	√	√
2160p30	√	(1)	√	√
2160p60	√	(2)	(2)	(2)
VGA 60Hz	DRU	DRU	DRU	DRU
SVGA 60Hz	DRU	DRU	DRU	√
XGA 60Hz	DRU	DRU	DRU	√
SXGA 60Hz	√	√	√	√
WXGA 60Hz	DRU	DRU	√	√
WXGA + 60Hz	DRU	√	√	√
UXGA 60Hz	√	√	√	(3)
WUXGA 60Hz	√	√	√	√
WSXGA 60Hz	√	√	√	√

注記 :

- これらのフォーマットは、NI-DRU と CPLL のいずれのサポート範囲にも含まれないため、CPLL ではサポートされません。
- このフォーマットは、HDMI 2.0 の最大ライン レートを超えるためサポートされません。
- このフォーマットは、送信についてはサポートされていますが、レシーバーでは現在のところサポートされていません。

HDMI トランスミッターおよびレシーバーに 2 つの種類のクワッド PLL (QPLL) を使用する場合は、VCO の動作範囲と設定可能な QPLL の通倍値は限られているため、ライン レートの制限が生じます。VPHY Controller コア ドライバーは、これらの制限を克服するために QPLL0 と QPLL1 を動的に切り替えます。トランスミッターについては、VPHY Controller コア ドライバーは、オーバーサンプリングと動的 QPLL スwitching を使用して、QPLL の制限を回避します。

QPLL0 の VCO は、9.8GHz ~ 16.3GHz の周波数範囲で動作しなければなりません。QPLL0 は、20、40、80、または 160 の通倍値を TMDS クロックに適用できます。



重要 : VCO の動作範囲と利用可能な QPLL0 の通倍値は限られているため、HDMI でサポートされるライン レートの範囲は不連続になります。

表 4 に、適用される QPLL0 の通倍値とサポートされる周波数範囲を TMDS クロック周波数帯ごとに示します。

表 4 : UltraScale GTH の QPLL0 の使用方法

TMDS クロック周波数 (MHz)	QPLL0 の通倍値	注記
<61.25	N/A	TX : オーバーサンプリング RX : NI-DRU が使用される
61.25 ~ 101.875	160	サポート
101.875 ~ 122.5	-	サポート対象外の TMDS クロック範囲
122.5 ~ 203.75	80	サポート

表 4 : UltraScale GTH の QPLL0 の使用方法 (続き)

TMDS クロック周波数 (MHz)	QPLL0 の通倍値	注記
203.75 ~ 245	-	サポート不可の TMDS クロック範囲
245 ~ 407	40	サポート
407 ~ 490	-	サポート不可の TMDS クロック範囲

クロックソースとして QPLL0 を使用する場合、TMDS クロック周波数が 101.875MHz ~ 122.5MHz、203.75MHz ~ 245MHz、および 407MHz を超えるビデオフォーマットは、通倍値を用いて有効な VCO 動作範囲を満たすことができないため、サポートされません。

QPLL1 の VCO は、8.0GHz ~ 13.0GHz の周波数範囲で動作しなければなりません。QPLL1 は、20、40、80、または 160 の通倍値を TMDS クロックに適用できます。



重要 : VCO の動作範囲と設定可能な QPLL1 の通倍値は限られているため、HDMI でサポートされるラインレートの範囲は不連続になります。

表 5 に、適用される QPLL1 の通倍値とサポートされる周波数範囲を TMDS クロック周波数帯ごとに示します。

クロックソースとして QPLL1 を使用する場合、TMDS クロック周波数が 81.25MHz ~ 100MHz、162.5MHz ~ 200MHz、および 325MHz ~ 400MHz のビデオフォーマットは、通倍値を用いて有効な VCO 動作範囲を満たすことができないため、サポートされません。

表 5 : UltraScale GTH の QPLL1 の使用方法 (HDMI RX)

TMDS クロック周波数 (MHz)	QPLL1 の通倍値	注記
<50.0	N/A	TX : オーバーサンプリング RX : NI-DRU が使用される
50.0 ~ 81.25	160	サポート
81.25 ~ 100	-	サポート不可の TMDS クロック範囲
100 ~ 162.5	80	サポート
162.5 ~ 200	-	サポート不可の TMDS クロック範囲
200 ~ 325	40	サポート
325 ~ 400	-	サポート不可の TMDS クロック範囲
400 ~ 650	20	サポート

表 6 : QPLL による RGB および YCbCr 4:4:4 ビデオフォーマットのサポート

解像度	ピクセルあたりのビット数			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	√	√	√
1080i50	√	√	√	√
1080i60	√	√	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	√	√	√	√
720p60	√	√	√	√
1080p24	√	√	√	√
1080p25	√	√	√	√

表 6 : QPLL による RGB および YCbCr 4:4:4 ビデオ フォーマットの サポート (続き)

解像度	ピクセルあたりのビット数			
	24	30	36	48
1080p30	√	√	√	√
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	√	√	√
2160p25	√	√	√	√
2160p30	√	√	√	√
2160p60	√	(1)	(1)	(1)
VGA 60Hz	√	DRU	DRU	DRU
SVGA 60Hz	√	DRU	DRU	√
XGA 60Hz	√	√	√	√
SXGA 60Hz	√	√	√	√
WXGA 60Hz	√	√	√	√
WXGA + 60Hz	√	√	√	√
UXGA 60Hz	√	√	√	(2)
WUXGA 60Hz	√	√	√	√
WSXGA 60Hz	√	√	√	√

注記 :

- このフォーマットは、HDMI 2.0 の最大ライン レートを超えるためサポートされません。
- このフォーマットは、送信についてはサポートされていますが、レシーバーでは現在のところサポートされていません。

HDMI Transmitter Subsystem

HDMI_TX_SS は、次の 3 つの AXI インターフェイスを備えています。

- ビデオ

ビデオ インターフェイスは AXI4-Stream スレーブ バスです。1 クロックあたり 1 ピクセル、2 ピクセル、または 4 ピクセルを伝達し、1 コンポーネントあたり 8、10、12、16 ビットをサポートします。このインターフェイスは、『AXI リファレンス ガイド』(UG761) [参照 6] の「ビデオ IP : AXI 機能の導入」で定義しているビデオプロトコルを使用します。

- オーディオ

オーディオ インターフェイスは 32 ビット AXI4-Stream スレーブ バスです。このサブシステムは、複数チャネルの非圧縮オーディオデータを伝送します。

- CPU

CPU インターフェイスは AXI4-Lite バス インターフェイスです。このインターフェイスを介して、MicroBlaze などのプロセッサはレジスタにアクセスしてサブシステムを制御します。この AXI4-Lite スレーブ インターフェイスは、シングルビート読み出し/書き込みデータ転送をサポートします。バースト転送はサポートしません。

- VPHY Controller コア ステータス側帯波

VPHY Controller コア ステータス側帯波インターフェイスは、リンク クロック ロック (ビット 0) およびビデオ クロック ロック (ビット 1) ステータス ビットを伝送する 2 ビット AXI4-Stream スレーブ バスです。

- 3 つの HDMI ストリーム

HDMI ストリーム インターフェイスは、VPHY Controller コアのトランスミッター側に接続される 40 ビット AXI4-Stream マスター バスです。

HDMI_TX_SS の内部ロジックは固定されており、修正や改変はできません。このサブシステムの内部ロジックに影響を与える、ユーザーが設定可能な 2 つのパラメーターがあります。これらのパラメーターは、AXI4-Stream 出力の 1 クロックあたりのピクセル数と、1 コンポーネントあたりの最大ビット数です。リファレンス デザインでは、これらのパラメーターはそれぞれ 2 と 12 に設定されます。

HDMI_TX_SS は、プロセッサ コアが CPU インターフェイスを介して設定する選択済みビデオ フォーマットに基づいて、ビデオ ストリームとオーディオ ストリームを HDMI ストリームに変換します。このサブシステムは、変換された HDMI ストリームを VPHY Controller コアに送信します。VPHY Controller コアは、このデータを、HDMI ケーブルを介して HDMI シンクに送信される電気信号に変換します。

詳細は、『HDMI 1.4/2/0 Transmitter Subsystem LogiCORE IP 製品ガイド』(PG235) [参照 3] を参照してください。

HDMI Receiver Subsystem

HDMI_RX_SS は、次の AXI インターフェイスを備えています。

- ビデオ

ビデオ インターフェイスは AXI4-Stream マスター バスです。1 クロックあたり 1 ピクセル、2 ピクセル、または 4 ピクセルを伝達し、1 コンポーネントあたり 8、10、12、16 ビットをサポートします。このインターフェイスは、『AXI リファレンス ガイド』(UG761) [参照 6] の「ビデオ IP : AXI 機能の導入」で定義しているビデオ プロトコルを使用します。

- オーディオ

オーディオ インターフェイスは 32 ビット AXI4-Stream マスター バスです。このサブシステムは、複数チャンネルの非圧縮オーディオ データを伝送します。

- CPU

CPU インターフェイスは AXI4-Lite バス インターフェイスです。このインターフェイスを介して、MicroBlaze などのプロセッサはレジスタにアクセスしてサブシステムを制御します。この AXI4-Lite スレーブ インターフェイスは、シングルビート読み出し/書き込みデータ転送をサポートします。バースト転送はサポートしません。

- VPHY Controller コア ステータス側帯波

VPHY Controller コア ステータス側帯波インターフェイスは、リンク クロック ロック (ビット 0) およびビデオ クロック ロック (ビット 1) ステータス ビットを伝達する 2 ビット AXI4-Stream スレーブ バスです。

- 3 つの HDMI ストリーム

HDMI ストリーム インターフェイスは、VPHY Controller コアからの復元されたデータを伝達する 40 ビット AXI4-Stream スレーブ バスです。

HDMI_RX_SS の内部ロジックは固定されており、修正や改変はできません。このサブシステムの内部ロジックに影響を与える、ユーザーが設定可能な 3 つのパラメーターがあります。これらのパラメーターは、AXI4-Stream 出力の 1 クロックあたりのピクセル数、1 コンポーネントあたりの最大ビット数、EDID RAM 容量です。リファレンス デザインでは、これらのパラメーターはそれぞれ 2、12、256 に設定されます。

HDMI_RX_SS は、VPHY Controller コアからの HDMI ストリームからビデオ ストリームとオーディオ ストリームを抽出します。

詳細は、『HDMI 1.4/2/0 Receiver Subsystem LogiCORE IP 製品ガイド』(PG236) [参照 4] を参照してください。

Utility Buffer

Utility Buffer コアは、MGT 差動入力バッファ (IBUFDS_GTE) として構成され、VPHY Controller コア クワッドの外部の MGT 基準クロック (すなわち、GTNORTHREFCLK0 または GTSOUTHREFCLK1) を使用できるようにします。

Video Test Pattern Generator

TPG コアは、ビデオシステムの立ち上げ、評価、デバッグ用のテストパターンを生成します。リファレンス デザインでは、このコアはパススルーモードとテストパターン生成モードの2つの主な動作モードを備えています。パススルーモードでは、入力 AXI4-Stream ビデオは変更されずに AXI4-Stream 出力インターフェイスに渡されます。生成モードでは、TPG コアは、デモ目的専用のカラーバーパターンを生成するように設定されます。



重要 : TPG コアは、ビデオシステムの色、品質、エッジ、動き性能のデバッグおよび評価用のさまざまなテストパターンを提供します。TPG コアは AXI4-Stream ビデオインターフェイスに挿入でき、システムビデオ信号をパススルーするか、テストパターンを挿入するかはユーザーが選択できます。

リファレンス デザインでは、TPG コアの次の機能を使用します。

- RGB、YUV 4:4:4、YUV 4:2:2、YUV 4:2:0
- 1 カラーコンポーネントあたり 8、10、12、16 ビットの入力および出力をサポート
- 64 x 64 から最大 4096 x 2160 までの空間解像度をサポート

詳細は、『ビデオテストパターンジェネレーター LogiCORE 製品ガイド』(PG103) [参照 7] を参照してください。

オーディオ用カスタム IP コア

リファレンス デザインには、オーディオデモ用の2つのカスタム IP コアが含まれています。これらの IP コアは `aud_pat_gen` と `hdmi_acr_ctrl` ですが、正式なザイリンクス コアではありません。これらは、HDMI_TX_SS および HDMI_RX_SS のオーディオ機能のデモ用としてのみ提供されます。ザイリンクスはこれらのコアをサポートしていません。オーディオ処理ソリューションを実装や供給はユーザーが行うものとされています。

ソフトウェアアプリケーション

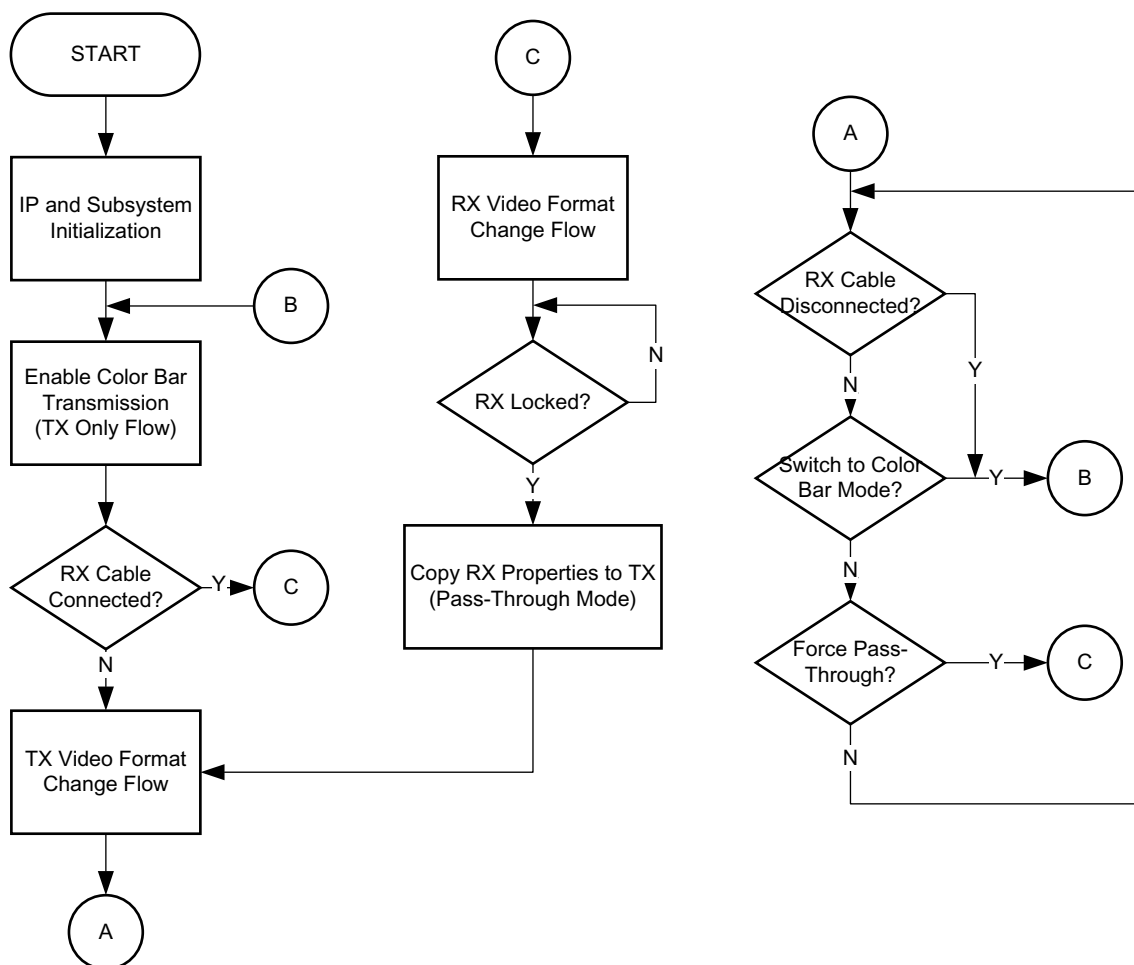
HDMI リファレンス デザインは、SDK の VPHY、HDMI_TX_SS、HDMI_RX_SS コアのビルトイン ドライバーを使用します。SDK は、含まれるサブコアの内部機能および管理を抽象化することで、すぐに使える形でのソリューションを提供します。VPHY、HDMI_TX_SS、HDMI_RX_SS の各ドライバーは、アプリケーション層の定義に従って複数の IP コアまたはサブシステムのインスタンスと連携して動作するベアメタルドライバーです。

VPHY Controller コアドライバーは、マルチギガビット トランシーバー モジュールおよびデジタルクロック マネージャー モジュールの動的リコンフィギュレーションを管理し、FPGA 物理インターフェイスとの間で HDMI ビデオのシームレスな送受信を可能にします。

HDMI_TX_SS サブシステム ドライバーと HDMI_RX_SS サブシステム ドライバーは、実行時の入力/出力ストリームの構成に基づいて、プロセッシング ユニットの介してデータ/制御フローを動的に管理します。内部では、これらのドライバーは、含まれるサブコア (レイヤー 1) ドライバーを利用して IP ハードウェアブロックをコンフィギュレーションします。レイヤー 2 (サブシステム) ドライバーは、各サブコアの機能セットを抽象化したビューを提供します。

サブシステム ドライバーはアクティブドライバーではなく、アプリケーション ソフトウェアを利用して、提供された API を用いてアクティブドライバーを設定します。アプリケーション ソフトウェアは、システムの外部入力を監視し、入力/出力ストリーム プロパティの変更を (提供された API を介して) サブシステムに伝達し、サブシステムの自動リコンフィギュレーション ルーチンをトリガーします。アプリケーション ソフトウェアは、HDMI TX アプリケーションと RX アプリケーションがボンディング モードになっている場合を除いて、これらのアプリケーションを個別に管理します。ボンディング モードでは、HDMI_TX_SS と HDMI_RX_SS は同じ GT PLL を共有します (詳細は、「リファレンス デザインに関するその他の情報」の「結果」を参照)。

図 3 にメインプログラムフローを示します。実行時に、ソフトウェアアプリケーションはいくつかの IP コアを初期化してアクティブにします。IP コアおよびサブシステムの初期化では、HDMI_TX_SS、HDMI_RX_SS、VPHY Controller コア IP のコンフィギュレーションの初期化、割り込みハンドラーの接続、API コールバック関数を実行します。次に、カラーバーの送信が開始されます。このプロセスは、TX ビデオフォーマット変更フローをトリガーするクロックジェネレーターの設定によって制限されます。



X15810-012216

図 3: メイン プログラム フロー

HDMI RX インターフェイス上でフル HDMI RX ケーブルが検出されると、ただちに RX ビデオフォーマット変更フローがトリガーされます。RX が入力ビデオフォーマットにロックされると、アプリケーションは HDMI RX ビデオプロパティを HDMI TX にコピーし、TX ビデオフォーマット変更フローでパススルーモードに移行します。

パススルーモードでは、HDMI RX ケーブルを外すか、または UART メニューでカラーバーを選択する (「C」を押す) ことにより、アプリケーションを強制的に TX 専用 (カラーバー) モードに移行させることができます。また、HDMI RX ケーブルを接続するか、またはアプリケーションで「P」を押すことにより、いつでも強制的にパススルーモードにできます。

TX ビデオフォーマット変更フロー

TX ビデオフォーマット変更フローは、VPHY Controller コアのクロック検出器モジュールからの割り込みを用い、TX TMDS クロック周波数を変更することでトリガーされます。VPHY Controller コアドライバーは、周波数イベントをデバウンスした後、伝送される新しいビデオフォーマットに基づいて GT TX のリコンフィギュレーションを開始します。GT のリコンフィギュレーションが完了すると、vid_phy_status_sb_tx_tdata の 0 ビットがアサートされます。

次に、VPHY Controller コアドライバーは、ビデオクロックと TX TMDS クロックの生成に使用される内部 TX デジタルクロックマネージャー (DCM) をコンフィギュレーションします。DCM のリコンフィギュレーションが完了すると、vid_phy_status_sb_tx_tdata の 1 ビットがアサートされます。

VPHY Controller コアのリコンフィギュレーションの完了後、HDMI_TX_SS のリコンフィギュレーションが実行されます。HDMI_TX_SS のリコンフィギュレーションはサブシステムドライバーによって処理されます。サブシステムドライバーがリコンフィギュレーションプロセスの実行を終了すると、アプリケーション内の TxStreamUpCallback が実行されます。

HDMI TX の処理全体を通して、VPHY Controller コア ドライバーと HDMI_TX_SS ドライバーはいくつかの API コールバックフックを実行します。ソフトウェア アプリケーション内の関数呼び出しに加えて、関数呼び出しを挿入または追加する場合は、必要に応じてこれらのコールバックを利用できます。

VPHY Controller コア ドライバーの TX コールバック

VPHY Controller コア ドライバーの TX コールバックには、次のものがあります。

- VPHY Controller コア HDMI TX Init コールバック

このコールバックは、VPHY Controller コア ドライバーでは XVPHY_HDMI_HANDLER_TXINIT という名前で、ソフトウェア アプリケーションで VphyHdmiTxInitCallback 関数にフックされます。これは、HDMI TX の TMDS クロック周波数が変更されるたびに実行または呼び出されます。ソフトウェア アプリケーションでは、フックの中で XV_HdmiTxSs_RefClockChangeInit HDMI_TX_SS API が呼び出され、HDMI_TX_SS の設定をリセットおよびクリアします。

- VPHY Controller コア HDMI TX Ready コールバック

このコールバックは、VPHY Controller コア ドライバーでは XVPHY_HDMI_HANDLER_TXREADY という名前で、ソフトウェア アプリケーションで VphyHdmiTxReadyCallback 関数にフックされます。これは、VPHY Controller コア ドライバーが TX ビデオ フォーマットの変更に必要な初期化ルーチンを完了するたびに実行または呼び出されます。フックには現在は空の関数が含まれています。

HDMI_TX_SS ドライバーのコールバック

HDMI_TX_SS ドライバーのコールバックには、次のものがあります。

- HDMI_TX_SS TX Connect コールバック

このコールバックは、HDMI_TX_SS ドライバーでは XV_HDMITXSS_HANDLER_CONNECT という名前で、ソフトウェア アプリケーションで TxConnectCallback 関数にフックされます。これは、HDMI TX ケーブルを接続または切断たびに実行または呼び出されます。TxConnectCallback ソフトウェア フックは、XV_HdmiTxSs_DetectHdmi20 API を介して、接続された HDMI シンクの HDMI 2.0 対応の有無を検出します。また、このソフトウェア フックは、ケーブルが接続された場合は差動入力クロック バッファを有効にし、ケーブルが切断された場合は無効にします。

- HDMI_TX_SS Vsync コールバック

このコールバックは、HDMI_TX_SS ドライバーでは XV_HDMITXSS_HANDLER_VS という名前で、ソフトウェア アプリケーションで TxVsCallback 関数にフックされます。これは、HDMI ストリームで垂直同期が検出されるたびに実行または呼び出されます。TxVsCallback ソフトウェア フックは、TX 専用モードでは NULL 補助情報フレームの送信を開始します。

- HDMI_TX_SS Stream Up コールバック

このコールバックは、HDMI_TX_SS ドライバーでは XV_HDMITXSS_HANDLER_STREAM_UP という名前で、ソフトウェア アプリケーションで TxStreamUpCallback 関数にフックされます。これは、HDMI_TX_SS がビデオ レディ (TX DCM Lock) 信号のアサートを検出したときに実行または呼び出されます。TxStreamUpCallback ソフトウェア フックは、オーバーサンプリング レートやビデオ プロパティの設定など、HDMI_TX_SS 内でビデオ ストリームの初期化を実行します。また、TX TMDS クロックを有効にし、FMC カードのリタイマー チップ内で TX ライン レートをプログラムします。

- HDMI_TX_SS Stream Down コールバック

このコールバックは、HDMI_TX_SS ドライバーでは XV_HDMITXSS_HANDLER_STREAM_DOWN という名前で、ソフトウェア アプリケーションでは使用されません。

RX ビデオ フォーマット 変更 フロー

RX ビデオ フォーマット 変更 フローは、VPHY Controller コア内のクロック検出器モジュールからの割り込みを用い、RX TMDS クロック 周波数を変更することでトリガーされます。VPHY Controller コア ドライバーは、周波数 イベントをデバウンスした後、伝送される新しいビデオ フォーマットに基づいて GT RX のリコンフィギュレーションを開始します。GT のリコンフィギュレーションが完了すると、vid_phy_status_sb_rx_tdata の 0 ビットがアサートされます。

HDMI_RX_SS リコンフィギュレーションは、vid_phy_status_sb_rx_tdata の 0 ビットのアサートを検出し、新しいビデオ フォーマット用の初期化を開始します。この初期化の中で XV_HDMIRX_STATE_STREAM_INIT コールバックが呼び出され、VPHY Controller コアの内部で RX DCM をコンフィギュレーションします。次に、VPHY Controller コア ドライバーは、ビデオ クロックと RX TMDS クロックの生成に使用される内部 RX DCM をコンフィギュレーションします。DCM のリコンフィギュレーションが完了すると、vid_phy_status_sb_rx_tdata の 1 ビットがアサートされます。HDMI_RX_SS はこのイベントを検出し、RxStreamUpCallback を呼び出してリコンフィギュレーション プロセスを完了します。

HDMI RX の処理全体を通して、VPHY Controller コア ドライバーと HDMI_RX_SS ドライバーはいくつかの API コールバック フックを実行します。ソフトウェア アプリケーション内の関数呼び出しに加えて、さ関数呼び出しを挿入または追加する場合は、必要に応じてこれらのコールバックを利用できます。

VPHY Controller コア ドライバーの RX コールバック

VPHY Controller コア ドライバーの RX コールバックには、次のものがあります。

- VPHY Controller コア HDMI RX Init コールバック

このコールバックは、VPHY Controller コア ドライバーでは XVPHY_HDMI_HANDLER_RXINIT という名前で、ソフトウェア アプリケーションで VphyHdmiRxInitCallback 関数にフックされます。これは、HDMI RX の TMDS クロック 周波数に変更されるたびに実行または呼び出されます。ソフトウェア アプリケーションでは、フックの中で XV_HdmiTxSs_RefClockChangeInit HDMI_RX_SS API が呼び出され、新しい TMDS クロック比を取得します。

- VPHY Controller コア HDMI RX Ready コールバック

このコールバックは、VPHY Controller コア ドライバーでは XVPHY_HDMI_HANDLER_RXREADY という名前で、ソフトウェア アプリケーションで VphyHdmiRxReadyCallback 関数にフックされます。これは、VPHY Controller コア ドライバーが RX ビデオ フォーマットの変更に必要な初期化ルーチンを完了するときに実行または呼び出されます。このフックは、HDMI_RX_SS 内のクロックおよびライン レート パラメーターを更新します。

HDMI_RX_SS ドライバーのコールバック

HDMI_RX_SS ドライバーのコールバックには、次のものがあります。

- HDMI_RX_SS RX Connect コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_CONNECT` という名前で、ソフトウェア アプリケーションで `RxConnectCallback` 関数にフックされます。これは、HDMI RX ケーブルを接続または切断されるたびに実行または呼び出されます。`RxConnectCallback` ソフトウェア フックは、ケーブルが接続された場合は差動入力クロック バッファを有効にし、ケーブルが切断された場合は無効にします。

- HDMI_RX_SS Auxiliary コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_AUX` という名前で、ソフトウェア アプリケーションで `RxAuxCallback` 関数にフックされます。これは、HDMI RX ストリームで補助パケットが検出されるたびに実行または呼び出されます。`RxAuxCallback` ソフトウェア フックは、パススルー モードでは補助情報フレームの再送信を開始します。

- HDMI_RX_SS Audio コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_AUD` という名前で、ソフトウェア アプリケーションで `RxAudCallback` 関数にフックされます。これは、HDMI RX ストリームでオーディオ パケットが検出されるたびに実行または呼び出されます。`RxAudCallback` ソフトウェア フックは、パススルー モードではオーディオパケットの再送信を開始します。

- HDMI_RX_SS Link Status コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_LNKSTA` という名前で、ソフトウェア アプリケーションで `RxLnkStaCallback` 関数にフックされます。これは、HDMI RX ストリームで変化が検出されるたびに実行または呼び出されます。`RxLnkStaCallback` ソフトウェア フックは、RX リンク エラー ステータスが 255 カウントに達したとき、(VPHY Controller コア内の) RX PLL リセット パルスを発行します。

- HDMI_RX_SS DDC コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMITXSS_HANDLER_DDC` という名前で、ソフトウェア アプリケーションでは使用されません。

- HDMI_RX_SS Stream Down コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_STREAM_DOWN` という名前で、ソフトウェア アプリケーションでは `IsPassThrough` フラグをクリアして TX 専用モードを開始するために使用されます。

- HDMI_RX_SS Stream Init コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_STREAM_INIT` という名前で、ソフトウェア アプリケーションでは入力 HDMI RX ストリームに基づいて VPHY Controller コア内で RX DCM のリコンフィギュレーションを開始するために使用されます。

- HDMI_RX_SS Stream Up コールバック

このコールバックは、HDMI_RX_SS ドライバーでは `XV_HDMIRXSS_HANDLER_STREAM_UP` という名前で、ソフトウェア アプリケーションで `RxStreamUpCallback` 関数にフックされます。これは、HDMI_RX_SS がビデオ レディ (RX DCM Lock) 信号のアサートを検出したときに実行または呼び出されます。`RxStreamUpCallback` ソフトウェア フックは、HDMI RX ストリームのパラメーターを HDMI_TX_SS にコピーして、パススルー モードを準備します。

ユーザー定義情報フレームの送信

ユーザー定義情報フレームは、HDMI_TX_SS ソフトウェア API を使用して送信できます。情報フレーム API 関数呼び出しを送信する前に、4 バイト ヘッダーと 32 バイト データ (ペイロード) から成る情報フレームのデータ構造が構築されている必要があります。

情報フレームのデータ構造へのアクセスと変更

情報フレームのデータ構造へのアクセスと変更には、一連のポインターによって `HDMI_TX_SS` 変数プレースホルダーを使用します。リファレンス デザインでは、`HDMI_TX_SS` 変数はアプリケーション (`xhdmi_example.c`) 内で `HdmiTxSs` として宣言されます。情報フレーム ヘッダーおよびデータのプレースホルダーは次の場所にあります。

- 情報フレーム ヘッダー (4 バイト)

```
HdmiTxSs.HdmiTxPtr->Aux->Header
```

- 情報フレーム データ (32 バイト)

```
HdmiTxSs.HdmiTxPtr->Aux->Data
```

情報フレームの送信

ヘッダーとデータの変更が完了したら、アプリケーション内で情報フレーム送信 API を呼び出すことができます。関数呼び出しの例を次に示します。

```
XV_HdmiTx_AuxSend(HdmiTxSs.HdmiTxPtr);
```

リファレンス デザインのツール フローおよび検証

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 7 に、リファレンス デザインで使用されるツール フローおよび検証手順を示します。

表 7: リファレンス デザインの詳細

パラメーター	説明
全般	
開発者	Gilbert Magnaye、Marco Groeneveld
ターゲット デバイス	Kintex UltraScale FPGA
ソース コードの提供	なし
ソース コードの形式 (提供される場合)	N/A
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、サードパーティ、Vivado Design Suite からデザインへのコード/IP の使用(使用した場合はその詳細)	Vivado IP インテグレーターからコアを生成
シミュレーション	
論理シミュレーションの実施	N/A
タイミングシミュレーションの実施	N/A
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用テストベンチの提供	N/A
テストベンチの形式	N/A
使用したシミュレータ/バージョン	N/A
SPICE/IBIS シミュレーションの実施	N/A
インプリメンテーション	
使用した合成ツール/バージョン	Vivado 2015.4 Synthesis
使用したインプリメンテーション ツール/バージョン	Vivado 2015.4 Implementation
スタティック タイミング解析の実施	あり
ハードウェア検証	
ハードウェア検証の実施	あり
使用したプラットフォーム	Xilinx UltraScale FPGA KCU105 ボード評価キット

必要な環境

ハードウェア要件

リファレンス システムのハードウェア要件は次のとおりです。

- Xilinx Kintex UltraScale FPGA KCU105 ボード評価キット
- inrevium HDMI 2.0 ドーター カード (TB-FMCH-HDMI4K)
- 2本の HDMI ケーブル
- HDMI 2.0 ビデオ ソース (例: DVD プレイヤー)
- HDMI 2.0 シンク (例: 超高精細 (UHD) TV セット)

ソフトウェア要件

リファレンス システムのソフトウェア要件は次のとおりです。

- Vivado Design Suite 2015.4
- ソフトウェア開発キット (SDK) 2015.4
- ソフトウェア ターミナル (例: Tera Term、ハイパーターミナル、PuTTY)

リファレンス デザイン ファイル

図 4 に、リファレンス デザインのディレクトリ構造を示します。

重要: リファレンス デザインはルート ディレクトリの近くに展開してください。

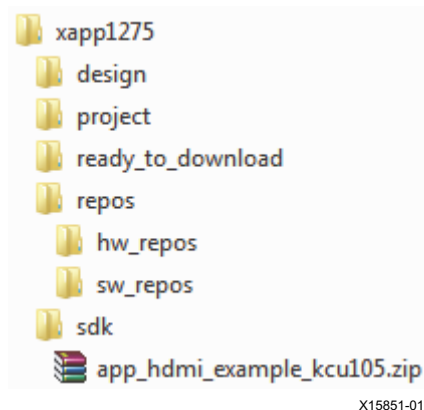


図 4: リファレンス デザインのディレクトリ構造

図 5 に、XAPP1275 のディレクトリ構造を示します。

```

all.tcl           Main TCL file to be executed in Vivado TCL command prompt
design            Contains the kcu105 specific design files such as IPI TCL, top
                level wrapper and XDC
| hdmi_example_kcu105.tcl:    TCL for constructing the HDMI IPI design
| hdmi_example_kcu105.xdc:    HDMI Reference Design top level constraint file

project          Vivado project and generated files container folder

ready_to_download  Contains the bitfile and the ELF of the system
| hdmi_example_kcu105_wrapper.bit  Reference design precompiled bitfile
| app_hdmi_example_kcu105.elf      SW application that is embedded in the bitfile

repos           Contains the HW and SW repositories
|
| \hw_repos      HW repository for custom IPs
| \sw_repos      SW repository (empty)

sdk             Contains the zip package of the reference design SW application

```

X15853-012216

図 5 : XAPP1275 のディレクトリ構造

ライセンス

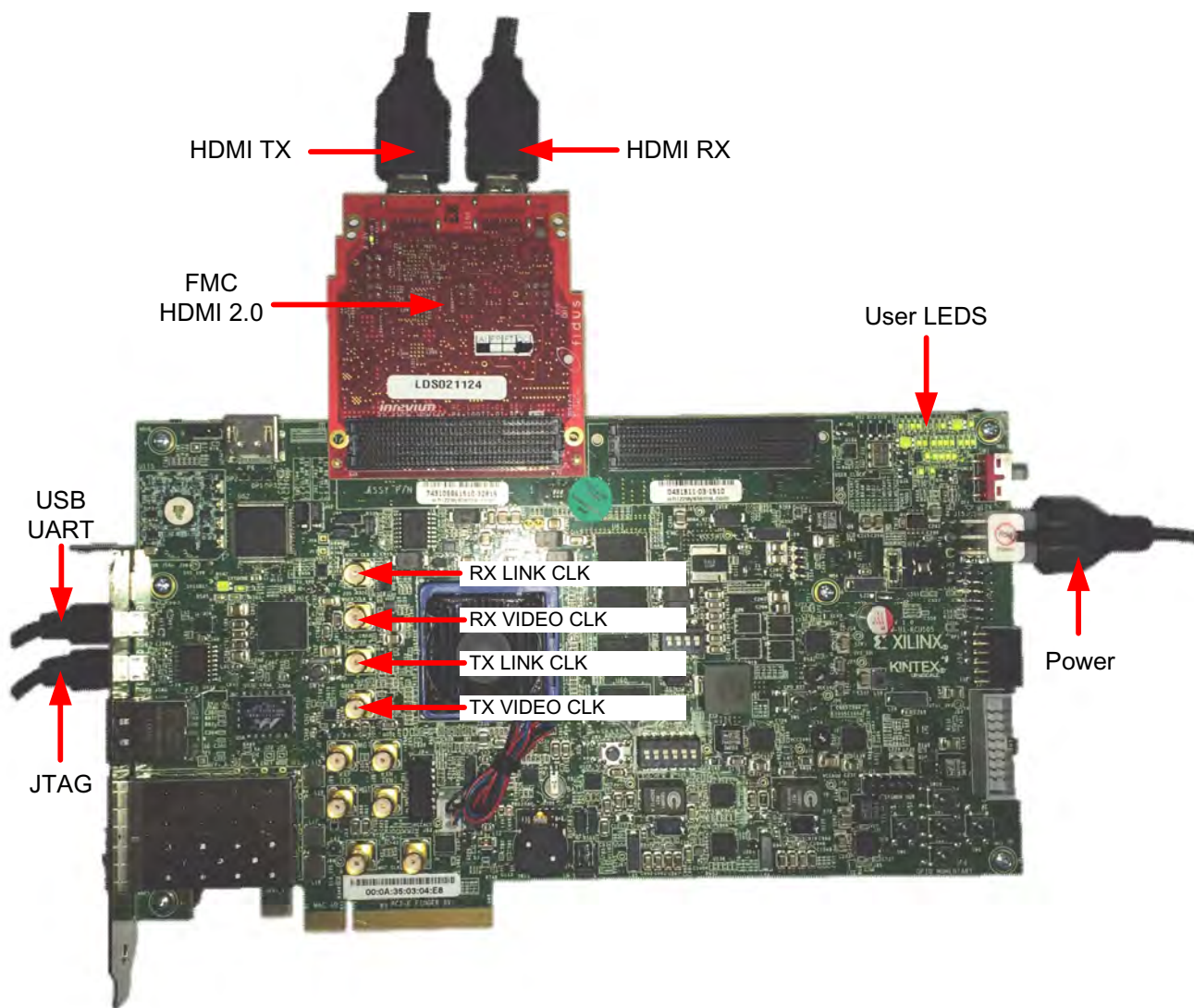
Video PHY Controller、HDMI 1.4/2.0 Transmitter Subsystem、HDMI 1.4/2.0 Receiver Subsystem、および TPG コアのライセンスがインストールされていることを確認してください。

リファレンス デザインの実行手順

このセクションでは、HDMI リファレンス デザインのセットアップ、実行、結果について説明します。

ボードのセットアップ

サンプル デザインは、Kintex UltraScale KCU105 開発ボードをターゲットとしています。サンプル デザインは、inrevium TB-FMCH-HDMI4K FMC カードをサポートしています。図 6 に、ボードのセットアップを示します。



X15811-012216

図 6: HDMI リファレンス デザインのセットアップ

ハードウェアのセットアップ手順

1. USB ケーブルをホスト PC から USB JTAG ポートに接続します。適切なデバイスドライバがインストールされていることを確認します。
2. 別の USB ケーブルをホスト PC から USB UART ポートに接続します。「ハードウェア要件」に記載された USB-UART ドライバがインストールされていることを確認します。FMCH 電源レールを設定するための FMCH VADJ の調整については、『KCU105 ボード ユーザー ガイド』(UG917) [参照 8] を参照してください。
3. TB-FMCH-HDMI4K ボードを KCU105 ボードの HPC FMC コネクタに接続します。
4. HDMI TX ポートを HDMI 1.4/2.0 シンクまたはモニター (超高精細 (UHD) TV セットなど) に接続します。
5. HDMI RX ポートを HDMI 1.4/2.0 ソース (DVD プレイヤーなど) に接続します。
6. KCU105 ボードを電源スロット J49 に接続します。
7. KCU105 ボードの電源をオンにします。
8. 次のように標準 COM ポートを設定したホスト PC 上でターミナルプログラム (ハイパーターミナルなど) をします。
 - 。 ボーレート : 115200
 - 。 データビット : 8

- 。 パリティ:なし
- 。 ストップ ビット:1
- 。 フロー制御:なし

表 8 に、このリファレンス デザインで使用されるオンボード LED を示します。詳細は、「デバッグ」を参照してください。

表 8: ユーザー LED

LED	説明
0	HDMI TX Subsystem ロック
1	使用しない
2	使用しない
3	使用しない
4	使用しない
5	使用しない
6	TX クロック ハートビート
7	RX クロック ハートビート

KCU105 ボードの FMCH VADJ の調整

KCU105 ボード システム コントローラーは、HDMI 2.0 FMC カード用の VADJ 電源レールに自動的に電力を供給しないため、ビット ストリームのコンフィギュレーション前に UART ターミナルで手作業で 1.8V に設定する必要があります。KCU105 ボードでは、micro USB コネクタを使用して、Zynq-7000 AP SoC システム コントローラーおよび UltraScale FPGA 用の UART にアクセスできます。

UART ターミナルに接続するには、次の手順に従います。

1. USB ケーブルを KCU105 ボードの USB UART コネクタから Windows 搭載 PC に接続します。
2. Windows デバイス マネージャーを使用して、Zynq-7000 AP SoC システム コントローラー用 UART に割り当てられた仮想 COM ポートと UltraScale FPGA 用 UART に割り当てられた仮想 COM ポートを確認します。デバイス マネージャー画面の COM ポートの一覧では、CP210x に関連付けられている拡張 COM ポートが KCU105 ボード システム コントローラーに接続されているポートで、標準 COM ポートが FPGA UART に接続されているポートです。
3. ターミナル ウィンドウ (115200、8、N、1) を開き、COM ポートを KCU105 ボード システム コントローラーと通信するポートに設定します。
4. UART ターミナルの接続後、KCU105 ボードの電源を一旦切って再度入れ、UART ターミナルのシステム コントローラー メニューを更新します。システム コントローラー メニューで次のオプションを選択します。
 - 4.Adjust FPGA Mezzanine Card (FMC) Settings
5. 次のメニューで、次の項目を選択します。
 - 4.Set FMC VADJ to 1.8V

(KCU105 ボードの電源スイッチの近くにある) DS19 LED が点灯していれば、VADJ 電源は正常な状態です。

リファレンス デザインの実行

このセクションでは、ready_for_download ディレクトリ内のファイルを使用したシステムの実行手順を説明します。

1. Xilinx Microprocessor Debugger ツールを起動します。


```
[スタート] → [すべてのプログラム] → [Xilinx Design Tools] → [Vivado 2015.4] → [SDK]
```
2. ザイリンクスのコマンド シェル ウィンドウで、ready_for_download ディレクトリに変更します。


```
cd <unzip_dir>/xapp1275/ready_for_download
```
3. ビットストリームを FPGA にダウンロードします。

```
XMD% fpga -f hdmi_example_kcu105_wrapper.bit bit
```

4. XMD コマンド プロンプトを終了します。

```
XMD% exit
```



重要: FPGA のコンフィギュレーションが完了すると、すぐにソフトウェア アプリケーションの実行が開始します。実行ファイル (.elf) は、コンフィギュレーションファイル (download.bit) に埋め込まれています。

結果

このセクションでは、リファレンス デザインの結果について説明します。

TX 専用モード

有効な HDMI ソースが HDMI RX ポートに接続されていない場合は、ビットストリームのダウンロードが完了した直後に TX 専用モードが開始されます。TX 専用モードのハイパーターミナル画面には、次の出力が表示されます。

```
-----
---  HDMI SS + VPhy Example v1.0  ---
---  (c) 2015 by Xilinx, Inc.      ---
-----

Build Nov 27 2015 - 13:59:13

-----

TX cable is connected

-----

Starting colorbar

-----

---  MAIN MENU  ---
-----

i - Info
    => Shows information about the HDMI RX stream, HDMI TX stream,
        GT transceivers and PLL settings.

c - Colorbar
    => Displays the colorbar on the source output.

r - Resolution
    => Change the video resolution of the colorbar.

f - Frame rate
    => Change the frame rate of the colorbar.

d - Color depth
    => Change the color depth of the colorbar.

s - Color space
    => Change the color space of the colorbar.

p - Pass-through
    => Passes the sink input to source output.

l - GT PLL layout
    => Select GT transceiver PLL layout.

z - GT log
```

```
=> Shows the GT transceiver log information.  
e - Edid  
    => Display and set edid.  
a - Audio  
    => Audio options.
```

```
Starting colorbar  
TX stream is up  
-----  
Colorbar :  
    Color Format:      RGB  
    Color Depth:      8  
    Pixels Per Clock:2  
    Mode:              Progressive  
    Frame Rate:        60Hz  
    Resolution:        1920x1080@60Hz  
    Pixel Clock:       148500000  
-----
```

図 7 に HDMI シンクに表示されるカラー バー パターンを示します。



X15812-012216

図 7: TX 専用モード - 1920x1080p60 (デフォルト) カラーバーパターン



重要: 図 7 では、HDMI 2.0 シンクが低い解像度を拡大表示するように設定されていないため、カラーバーは 4K ディスプレイ全体を占めていません。

パススルー モード

有効な HDMI ソースが HDMI RX ポートに接続されている場合は、ビットストリームのダウンロードが完了した直後にパススルー モードが開始されます。HDMI シンクの表示は、HDMI RX ポートに接続されている HDMI ソースの内容によります。パススルー モードのハイパーターミナル画面には、次の出力が表示されます。

```
-----
---  HDMI SS + VPhy Example v1.0  ---
---  (c) 2015 by Xilinx, Inc.      ---
-----

Build Nov 27 2015 - 13:59:13

-----

TX cable is connected
RX cable is connected

-----

Starting colorbar

-----

---  MAIN MENU  ---
-----

i - Info
    => Shows information about the HDMI RX stream, HDMI TX stream,
        GT transceivers and PLL settings.
c - Colorbar
    => Displays the colorbar on the source output.
r - Resolution
    => Change the video resolution of the colorbar.
f - Frame rate
    => Change the frame rate of the colorbar.
d - Color depth
    => Change the color depth of the colorbar.
s - Color space
    => Change the color space of the colorbar.
p - Pass-through
    => Passes the sink input to source output.
l - GT PLL layout
    => Select GT transceiver PLL layout.
z - GT log
    => Shows the GT transceiver log information.
e - Edid
    => Display and set edid.
a - Audio
    => Audio options.

Starting colorbar
Active audio channels 2
RX stream is up
```

```

TX stream is up
-----
Pass-Through :
    Color Format:      RGB
    Color Depth:      8
    Pixels Per Clock: 2
    Mode:              Progressive
    Frame Rate:        60Hz
    Resolution:        1920x1080@60Hz
    Pixel Clock:       148500000
-----

```

リファレンス デザインに関するその他の情報

EDID (Extended Display Identification Data) メニューには、次の選択肢が含まれます。

- 接続されているシンクの EDID を表示する
- シンクの EDID を HDMI_RX_SS にコピーする -> コピーされた EDID 値が HDMI ソースによってフェッチされるように、「パススルーを強制する」必要があります。
- HDMI_RX_SS のデフォルト EDID を読み込む

ハードウェアの構築

このセクションでは、ハードウェア デザインの再構築について説明します。プロジェクトを再構築する前に、Video PHY Controller コア、HDMI 1.4/2.0 Transmitter Subsystem、HDMI 1.4/2.0 Receiver Subsystem、および TPG コアのライセンスがインストールされていることを確認してください。



重要 : Windows を使用する場合、ファイルパスが長すぎるためにコンパイルエラーが発生しないように、プロジェクトファイルはできるだけルート ディレクトリの近くに展開してください。たとえば、一般的な Windows システムへのインストールでは、C:\ にファイルを展開します。

Vivado Design Suite 2015.4 でのプログラム ファイルの生成

Vivado Design Suite 2015.4 でプログラム ファイルを生成するには、次の手順に従います。

1. Vivado Design Suite を起動します。
2. Tcl コンソールで、ワークスペースディレクトリに変更します。


```
> cd <unzip dir>\xapp1275
```
3. all.tcl スクリプトを実行し、プロジェクト ビットストリームの作成、コンパイル、生成を実行します。


```
> source all.tcl
```

SDK でのソフトウェアのコンパイル

SDK でソフトウェアをコンパイルするには、次の手順に従います。

OS の手順

1. 同じフォルダーの <unzip dir>\xapp1275\sdk に zip パッケージを展開します。

Vivado Design Suite の手順

- ビットストリーム生成の完了後、次のようにクリックして、Vivado Design Suite 内のハードウェア デザイン ファイル (HDF) をエクスポートします。

[File] → [Export] → [Export Hardware]

- [Include bitstream] をオンにします。
- HDF をエクスポートするパスを指定します。

<unzip dir>\xapp1275\sdk

- 次のようにクリックして、SDK を起動します。

[File] → [Launch SDK]

- [Launch SDK] ポップアップ ウィンドウで、ファイルパスに次のように入力します。

<unzip dir>\xapp1275\sdk

- [OK] をクリックします。ここで Vivado Design Suite は SDK を起動し、ハードウェア プロジェクトを自動的に作成します。

SDK ツールの手順

- ボード サポート パッケージ (BSP) プロジェクトを作成します。

[File] → [New] → [Board Support Package]

- BSP プロジェクト名 (bsp_hdmi_example_kcu105 など) を入力し、[Finish] をクリックします。
- [Board Support Package Settings] で [OK] をクリックします。BSP が生成されるのを待ちます。
- [File] → [Import] → [General] → [Existing Projects] をクリックして、ソフトウェア アプリケーションをワークスペースにインポートします。
- [Next] をクリックして、[Browse] をクリックします。
- 次のパスを参照します。

<unzip dir>\xapp1275\sdk

- [OK] をクリックします。
- app_hdmi_example_kcu105 が選択されていることを確認します。
- [Finish] をクリックします。
- インポートされたソフトウェア アプリケーションが bsp_hdmi_example_kcu105 を参照していることを確認します。ソフトウェア アプリケーションプロジェクトを右クリックして、[Change Referenced BSP] をクリックし、[bsp_hdmi_example_kcu105] をクリックします。[OK] をクリックします。

ソフトウェア アプリケーションがコンパイルされます。この処理には 2 分から 5 分程度かかります。これで、SDK 内で既存ソフトウェア アプリケーションの変更や新規ソフトウェア アプリケーションの作成が可能になります。

デバッグ

次のユーザー LED を使用して、リファレンス デザインのステータスを視覚的に確認できます。

- LED0

HDMI TX Subsystem ロック。この LED が消えている場合は、HDMI TX ストリームのビデオ タイミング情報が HDML_TX_SS の期待しているタイミングと一致しないことを意味します。HDMI TX ストリームのタイミングが適切であることを確認してください。あるいは、TX ケーブルを一旦切断してから再接続し、HDMI TX を再初期化してください。

- LED6

TX クロック ハートビート。この LED が点滅していない場合は、TX リンク クロックがアクティブになっていないことを意味します。TX ケーブルが接続され、有効な HDMI ストリームが HDMI TX ポートに入力されていることを確認してください。

- LED7

RX クロック ハートビート。この LED が点滅していない場合は、RX リンク クロックがアクティブになっていないことを意味します。RX ケーブルが接続されていることを確認してください。

既知の問題

- 10BPC で 720p60 の輝度および彩度が適切にならない。
- 10BPC で WXGA+ (1366x768p60) の輝度および彩度が適切にならない。
- CPLL が 10BPC で 2160p24/25/30 を送受信できない。
- TPG と HDMI_TX_SS 間で AXI4-Stream ビデオの互換性に問題があるため、送信専用モードで YUV420 色空間のビデオが出力されない。この制限に対処するには、TPG と HDMI_TX_SS 間に AXIS Video Remapper (2016 年 1 月リリース予定) を使用する必要があります。
- TPG と HDMI_TX_SS 間で AXI4-Stream ビデオの互換性に問題があるため、送信専用モードで PAL および NTSC ビデオフォーマットのビデオが出力されない。この制限に対処するには、TPG と HDMI_TX_SS 間に AXIS Video Remapper (2016 年 1 月リリース予定) を使用する必要があります。

参考資料

このアプリケーション ノートの参考資料は次のとおりです。

1. [ザイリンクス Kintex UltraScale FPGA KCU105 ボード評価キット](#) のウェブサイト
2. [inrevium TB-FMCH-3GSDI2A](#) のウェブサイト
3. 『HDMI 1.4/2.0 Transmitter Subsystem v1.0 LogiCORE IP 製品ガイド』([PG235](#))
4. 『HDMI 1.4/2.0 Receiver Subsystem v1.0 LogiCORE IP 製品ガイド』([PG236](#))
5. 『Video PHY Controller v2.0 LogiCORE IP 製品ガイド』([PG230](#))
6. 『AXI リファレンス ガイド』([UG761](#))
7. 『LogiCORE IP Test Pattern Generator 製品ガイド』([PG103](#))
8. 『KCU105 ボード ユーザー ガイド』([UG917](#))

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	内容
2016 年 1 月 27 日	1.0	初版

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、<http://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。
<http://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

ザイリンクスの製品は、フェイルセーフとして設計されたり意図されてはおらず、また、フェイルセーフの動作を要求するアプリケーション(具体的には、(I)エアバッグの展開、(II)車のコントロール(フェイルセーフまたは余剰性の機能(余剰性を実行するためのザイリンクスの装置にソフトウェアを使用することは含まれません)および操作者がミスをした際の警告信号がある場合を除きます)、(III)死亡や身体傷害を導く使用、に関するアプリケーション)を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。顧客は、そのよう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を単独で負います。

© Copyright 2016 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 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。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。