

HDCP 2.2 v1.0

製品ガイド

Vivado Design Suite

PG249 2016 年 10 月 5 日

この資料は表記のバージョンの英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。資料によっては英語版の更新に対応していないものがあります。日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

目次

IP の概要

第 1 章: 概要

HDMI 上での HDCP 2.2.....	4
アプリケーション	6
ライセンスおよび注文情報	6

第 2 章: 製品仕様

IP コア	8
規格	10
性能とリソース使用状況	10
ポートの説明	10

第 3 章: IP を使用するデザイン

一般的なデザイン ガイドライン	14
クロッキング	24
リセット	24

第 4 章: デザイン フローの手順

IP のカスタマイズおよび生成	25
IP への制約	27
シミュレーション	28
合成およびインプリメンテーション	28

付録 A: アプリケーション ソフトウェア開発

デバイスドライバ	29
----------------	----

付録 B: アップグレード

付録 C: デバッグ

ザイリンクス ウェブサイト	37
デバッグ ツール	38
ハードウェア デバッグ	38

付録 D: その他のリソースおよび法的通知

ザイリンクス リソース	40
参考資料	40
改訂履歴	41
お読みください: 重要な法的通知	41

はじめに

ザイリンクス HDCP (High-bandwidth Digital Content Protection) 2.2 トランスミッターおよびレシーバーは、HDCP 2.2 で保護されたトランスミッターから HDCP 2.2 準拠のダウンストリームレシーバーへオーディオ/ビジュアルコンテンツをセキュアに送信する方法を定義した HDCP 2.2 仕様をインプリメントします。HDCP 2.2 は HDCP 1.4 プロトコルを置き換えるもので、下位互換性はありません。一般に、HDCP 2.2 は UHD (Ultra-High Definition) 解像度のコンテンツを暗号化する場合に使用し、HDCP 1.4 はそれよりも低解像度のレガシコンテンツを保護する場合に使用します。この IP は HDCP 2.2 から HDMI へのマップをサポートしています。この IP はスタンドアロンコアではなく、HDMI Transmitter/Receiver Subsystem の一部として統合されています。

機能

- HDMI 上で動作する HDCP 2.2 トランスミッター/レシーバー
- HDCP 2.2 認証およびキー交換
- ビデオ、オーディオ、およびデータ アイランド パケットの暗号化/復号化
- 認証に必要な RSA 復号化のアクセラレーション
- 最大 2160p (UHD)、60fps の解像度をサポート
- リピーターおよびコンバーターをサポート
- SRM (System Renewability Message) および不正機器の排除

この IP について	
コアの概要	
サポートされるデバイスファミリ ⁽¹⁾	UltraScale+™ (Zynq® UltraScale+ を含む) UltraScale™ Zynq-7000 7 シリーズ
サポートされるユーザーインターフェイス	AXI4-Lite、AXI4-Stream
リソース	性能とリソース使用状況 (TX) 性能とリソース使用状況 (RX)
コアに含まれるもの	
デザインファイル	暗号化済み RTL
サンプルデザイン	なし
テストベンチ	なし
制約ファイル	XDC
シミュレーションモデル	なし
サポートされるソフトウェアドライバー ⁽²⁾	スタンドアロン
テスト済みデザインフロー ⁽³⁾	
デザイン入力	Vivado® Design Suite
シミュレーション	サポートされるシミュレータについては、 『Vivado Design Suite ユーザーガイド: リリースノートガイド、インストールおよびライセンス』 を参照
合成	Vivado 合成
サポート	
ザイリンクス サポート ウェブ ページ で提供	

注記:

1. サポートされているデバイスの一覧は、Vivado IP カタログを参照してください。
2. スタンドアロンドライバーの詳細は、SDK ディレクトリ (<install_directory>/SDK/<release>/data/embeddedsw/doc/xilinx_drivers.htm) を参照してください。Linux OS およびドライバーサポートの情報は、[ザイリンクス Wiki ページ](#)を参照してください。
3. サポートされているツールのバージョンは、[『Vivado Design Suite ユーザーガイド: リリースノートガイド、インストールおよびライセンス』](#)を参照してください。

概要

ザイリンクス HDCP 2.2 ソリューションは次のもので構成されます。

- HDCP 2.2 トランスミッター
- HDCP 2.2 レシーバー

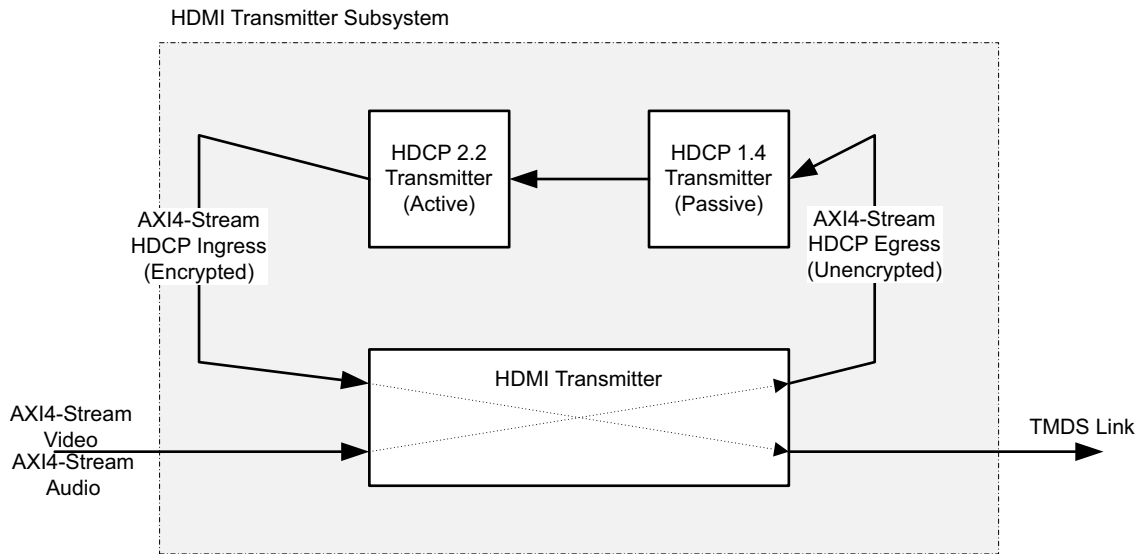
HDCP 2.2 トランスミッターおよびレシーバーは、ザイリンクス HDMI Transmitter/Receiver Subsystem (詳細は『HDMI 1.4/2.0 Transmitter Subsystem v2.0 製品ガイド』(PG235) [参照 1] および『HDMI 1.4/2.0 Receiver Subsystem v2.0 製品ガイド』(PG236) [参照 2] を参照) の一部としてのみインスタンシエートできます。以降のセクションでは、HDMI Transmitter/Receiver Subsystem 内で HDCP 2.2 を使用する方法的概要を説明します。

HDMI 上での HDCP 2.2

HDCP は、ザイリンクス HDMI Transmitter/Receiver Subsystem のコンフィギュレーションで次のいずれかに設定できます。

- HDCP なし
- HDCP 1.4 のみ
- HDCP 2.2 のみ
- HDCP 1.4 および HDCP 2.2

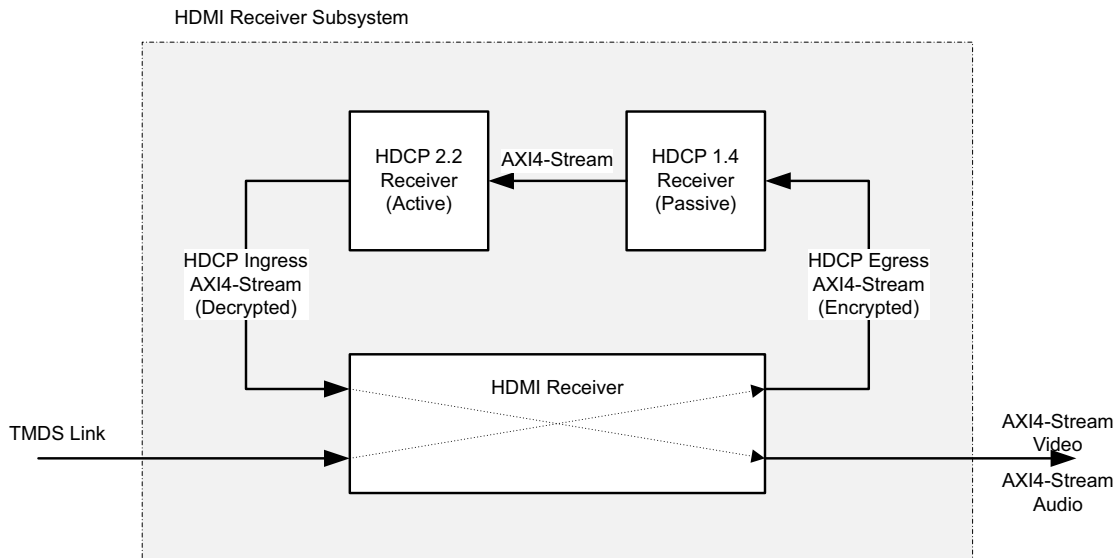
図 1-1 に、HDCP 1.4 と HDCP 2.2 を両方有効にした場合の HDMI Transmitter Subsystem の構成を示します。両方の HDCP プロトコルを有効にした場合、HDMI Subsystem は HDCP 1.4 と HDCP 2.2 を数珠つなぎにしたカスケード接続トポロジとして自身をコンフィギュレーションします。HDMI Transmitter の HDCP Egress インターフェイスからは暗号化されていない A/V データが送出され、これをアクティブな HDCP ブロックで暗号化した後、HDCP Ingress インターフェイス経由で HDMI Transmitter に戻してからリンクに送信します。HDMI Transmitter Subsystem は片方の HDCP プロトコルをパッシブにして、両方の HDCP プロトコルが同時にアクティブにならないようにします。



X16252-091616

図 1-1: HDMI Transmitter と HDCP 2.2 の組み合わせ

同様の方法で、HDMI Receiver は HDCP 1.4 または 2.2 トランスミッターからリンク経由で受信したデータを復号化できます (図 1-2)。HDMI Transmitter/Receiver Subsystem のカスタマイズでは、必要な HDCP プロトコルのみを有効にする必要があります。HDCP 2.2 は任意の解像度のビデオの暗号化に使用できますが、使用する暗号方式は HDCP レシーバーがサポートする解像度に応じて HDCP トランスミッターが選択します。HDCP トランスミッターは、たとえば 2160p60Hz 未満の解像度では HDCP 1.4 を選択し、2160p60Hz 以上の解像度では HDCP 2.2 を選択できます。HDMI Subsystem でどちらのプロトコルを有効にするかは、サポートされるユースケースを考慮してアプリケーションごとに決定する必要があります。



X16253-091616

図 1-2: HDMI Receiver と HDCP 2.2 の組み合わせ

アプリケーション

ザイリンクス HDCP 2.2 のトランスミッター / レシーバーはザイリンクス HDMI Transmitter/Receiver Subsystem にシームレスに統合されているため、ユーザー アプリケーションに簡単に統合でき、開発期間を短縮できます。ザイリンクス HDCP 2.2 ソリューションはトランスミッターとレシーバー、およびソフトウェア ドライバーで構成されます。解像度は最大 2160p60Hz (UHD) までサポートされます。

HDCP 2.2 トランスミッター / レシーバーは AES-128 暗号化を実装しているほか、認証およびキー交換プロセスの一部で使用する乱数生成や RSA 復号化アクセラレーションもサポートしています。認証およびキー交換プロセスで使用する乱数には、ハードウェア乱数生成器 (RNG) から得られる真性乱数を使用します。真性乱数は一般にアルゴリズムや数学モデルによって生成される擬似乱数とは異なり、温度変動やプロセスのばらつきなど実世界の現象に基づいて生成されます。HDCP 2.2 レシーバーにはマスター キーの復号化に必要な RSA べき剰余演算を高速化するモンゴメリ乗算器 (MMULT) が実装されています。

ライセンスおよび注文情報

ライセンス チェッカー

IP にライセンス キーが必要な場合、そのキーの認証が必要です。Vivado® デザイン ツールでは、設計フローにライセンスが必要な IP の使用をゲーティングする、ライセンス チェックポイントが複数あります。ライセンス チェックが正常に終了すると、IP の生成が継続されます。正常に終了しなければ、IP の生成はエラーとなり停止します。ライセンス チェックポイントが適用されるのは、次のツールです。

- Vivado 合成
- Vivado インプリメンテーション
- write_bitstream (Tcl コマンド)



重要: チェックポイントでは、IP のライセンス レベルは無視されます。有効なライセンスの有無のみを検証します。IP ライセンス レベルは確認しません。

ライセンスの種類

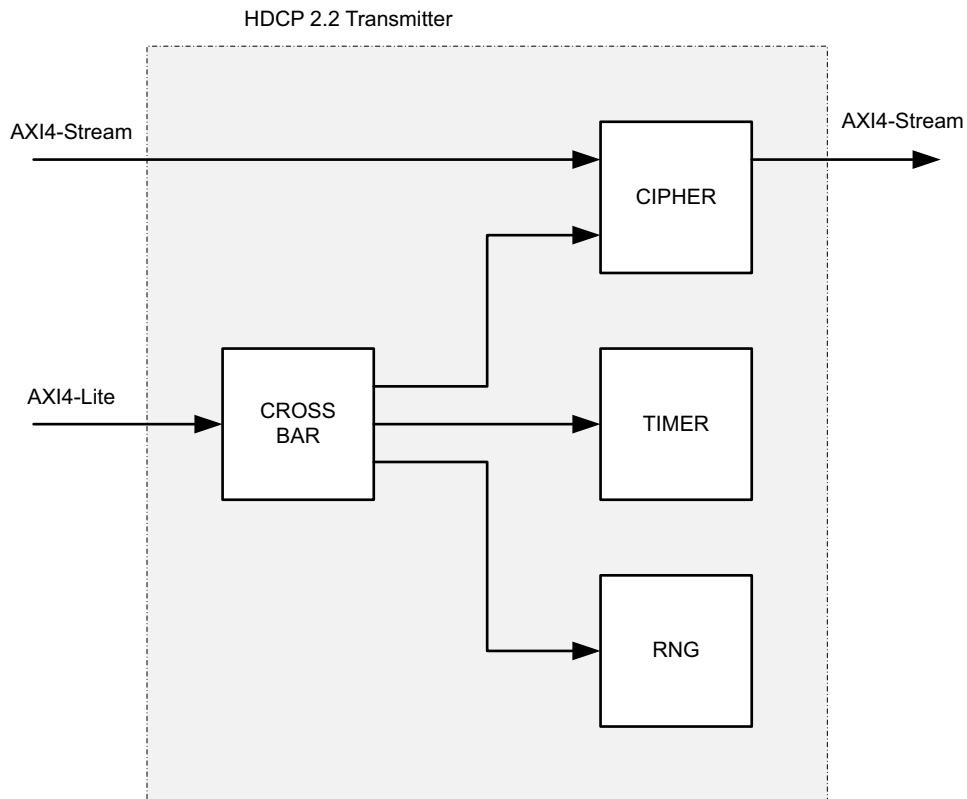
このザイリンクス IP モジュールは、[ザイリンクス コア ライセンス契約](#)に基づいて提供されます。このモジュールは、Vivado® Design Suite に付属します。シミュレーションおよびハードウェアでコアのすべての機能を利用するには、コアのライセンスをご購入いただく必要があります。価格および提供状況については、[ザイリンクス販売代理店](#)にお問い合わせください。

その他のザイリンクス LogiCORE IP モジュールに関する情報は、[ザイリンクス IP コア](#)のページを参照してください。その他のザイリンクス LogiCORE IP モジュールおよびツールの価格や提供状況については、[ザイリンクス販売代理店](#)にお問い合わせください。

この IP/リファレンス デザインを使用すると、ザイリンクス LogiCORE HDMI (High-Definition Multimedia Interface) IP ソリューションを使用したザイリンクスのシリコン デバイスに HDCP (High-bandwidth Digital Content Protection) の暗号化と認証の機能を実装できます。リファレンス デザインには、外部ストレージ デバイスから HDCP デバイス キーを安全に読み出してザイリンクスのシリコン デバイスに格納する際に使用できるロジック ブロックも含まれますが、これを使用するか同様の機能を実行するロジックを独自に開発するかはユーザーが選択できます。HDCP デバイス キーはリファレンス デザインには付属しておらず、いかなる場合にもザイリンクスから提供されることはありません。この IP およびリファレンス デザインを使用して HDCP を実装する場合は、HDCP Adopter の会員資格を取得して Digital Content Protection, LLC (DCP) から直接デバイス キーを入手する必要があります。この方法で HDCP デバイス キーを入手しない限り、この IP およびリファレンス デザインを使用して HDCP を製品に正しく実装することはできません。

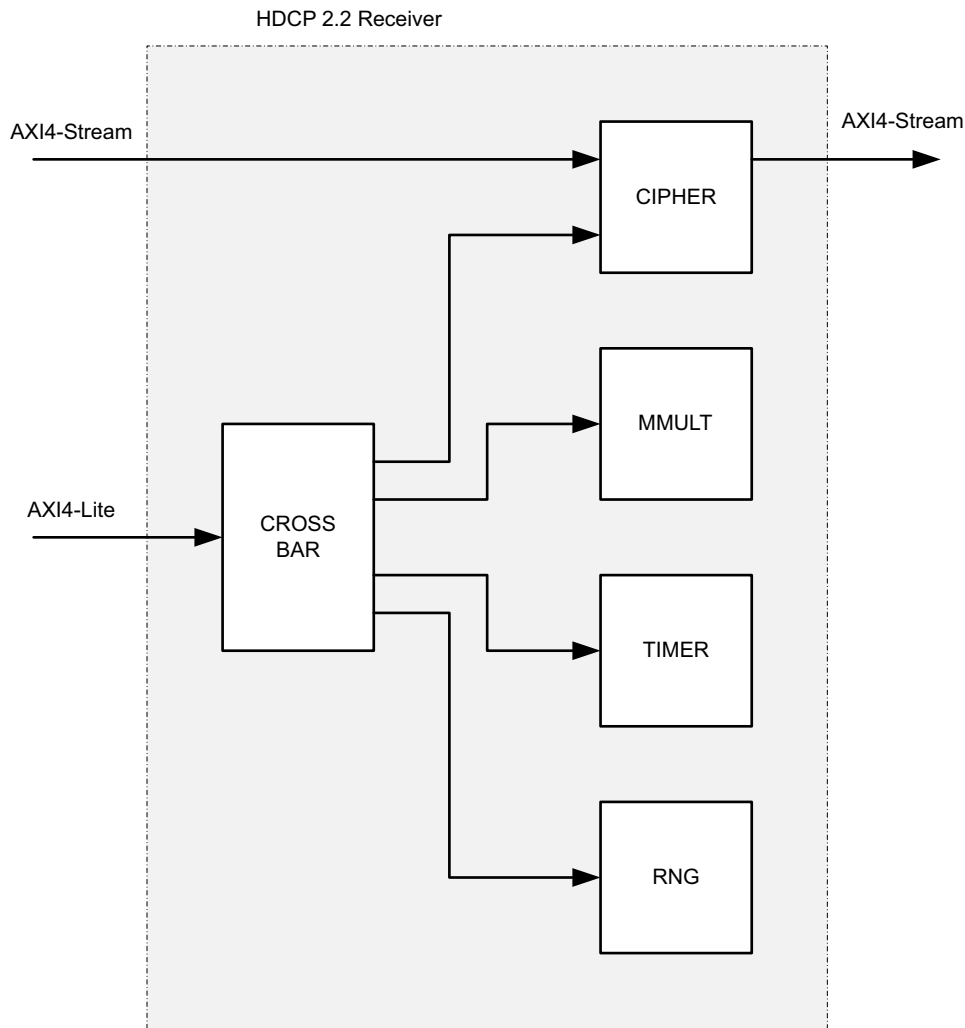
製品仕様

HDCP 2.2 プロトコルを使用すると、HDCP 2.2 対応トランスミッターとレシーバーの間でオーディオ/ビジュアルコンテンツをセキュアに送信できます。このプロトコルは認証およびキー交換 (AKE) とオーディオ/ビジュアルコンテンツの暗号化/復号化という 2 つの異なる要素で構成されます。AKE の役割は、プロトコル仕様に準拠したステートマシンを実装することにより、HDCP セッションを確立および維持することにあります。認証およびキー交換に成功すると、トランスミッター/レシーバーはコンテンツを暗号化/復号化できるようになります。図 2-1 と図 2-2 に、HDCP 2.2 トランスミッターおよびレシーバーのブロック図を示します。次のセクションでは HDCP 2.2 トランスミッター/レシーバー サブシステムを構成する IP コアについて説明します。



X16254-091616

図 2-1: HDCP 2.2 トランスミッターのブロック図



X16255-091616

図 2-2: HDCP 2.2 レシーバーのブロック図

IP コア

HDCP 2.2 トランスミッターおよびレシーバーは、次に説明する IP コアで構成されます。

Cipher

Cipher は AXI4-Stream インターフェイス経由でオーディオ/ビジュアル コンテンツを受信し、暗号化と復号化を実行します。Cipher が AES-128 ブロック暗号を使用してフレームをいつ暗号化または復号化するかは、HDMI サブシステムから提供される ESS (Encryption Status Signaling) に基づいて判断します。

HDCP 2.2 Cipher は HDCP 2.2 プロトコルの要件に従い、AES (Advanced Encryption Standard) ベースの暗号実装 (ブロック サイズ 128 ビット) を使用してオーディオ/ビジュアル コンテンツを暗号化/復号化します。Cipher はセッション キー (k_s) と DCP グローバル定数 (lc_{128}) の 2 つの秘密パラメーターを使用して暗号化を実行します。 lc_{128} は Digital Content Protection (DCP) から HDCP 2.2 Adopter に対して発行される定数で、 k_s は認証およびキー交換プロセス中に生成される乱数値です。いずれのパラメーターも、最初のフレームが暗号化または復号化される前に Cipher にロードされます。

HDMI で HDCP 2.2 を使用する場合、Cipher はカウンター モードで動作し、フレームおよびデータワードに基づいてカウントがインクリメントします。受信したフレームをいつ復号化するかは、TMDS チャンネルで送信される ESS (Encryption Status Signaling) 制御文字の情報によりレシーバーに通知されます。HDCP 2.2 トランスミッターの Cipher は ESS を生成し、HDCP 2.2 レシーバーの Cipher は ESS に応答します。

Random Number Generator (RNG)

Random Number Generator は、認証およびキー交換プロトコルで使用する真性乱数 (すなわち k_m 、 r_{tx} 、 r_{rx} 、 r_n 、 k_s) の生成に必要です。

HDCP 2.2 プロトコルでは、トランスミッターとレシーバーの間で交換する乱数を認証およびキー交換中に生成することが要求されています。これら乱数の一部は擬似乱数でもかまいませんが、マスター キー (k_m) やセッション キー (k_s) などの秘密キーには真性乱数を使用する必要があります。HDCP 2.2 Random Number Generator (RNG) は温度、電圧変動、およびシステム ノイズの影響を受けやすいリング オシレーターの発振周波数をエントロピー源として利用して真性乱数を生成します。

Montgomery Modular Multiplier (MMULT)

モンゴメリ剰余乗算は HDCP 2.2 レシーバーでのみ使用します。これは、プロトコルのタイミング要件を満たすために PKCS#1 のべき剰余演算をハードウェアにオフロードするために必要です。

認証およびキー交換中のマスター キー (k_m) 交換は、RSA アルゴリズムに基づく公開キー暗号システム (PKCS #1) を使用してセキュアに実行されます。トランスミッターが公開キーを使用して k_m を暗号化する際のタイミング制約は、HDCP 2.2 プロトコルでは規定されていません。さらに、公開キーの公開指数はそれほど大きくないことから、RSA 暗号化プリミティブ (RSAEP) はソフトウェアで実装できます。

これに対し、RSA 秘密キーを使用したマスター キー ($E_{k_{pub}}(k_m)$) 復号化については HDCP 2.2 プロトコルで 1 秒のタイミング制約が規定されています。また、秘密指数は値が大きく大量の演算が必要となるため、ソフトウェアのみで実装するとエンベデッド アプリケーションで 1 秒のタイミング要件を満たすのは困難です。このため、RSA 復号化プリミティブ (RSADP) は Montgomery Modular Multiplier (MMULT) およびバイナリ法 (Square and Multiply) を使用して一部をハードウェアへオフロードします。この方式により、マスター キーの復号化にかかる時間を大幅に短縮しています。

Timer

これは AXI Timer LogiCORE IP で、2 つの内部カウンターを使用するようにコンフィギュレーションします。カウンターの 1 つはウォッチドッグ タイマーとして使用し、タイムアウトになると割り込みを生成します。もう 1 つはログを記録する際のタイムスタンプ生成に使用します。ウォッチドッグ タイマーをソフトウェア ドライバーでトリガーすることで、メッセージ トランザクション間のタイムアウト期間を追跡できます。

HDCP 2.2 認証およびキー交換プロセス中、トランスミッターとレシーバーはウォッチドッグ タイマーを使用してメッセージ トランザクションの時間間隔を計測する必要があります。メッセージ トランザクションがタイミング要件に違反した場合、認証およびキー交換プロセスは中止されます。HDCP 2.2 Timer に必要なタイムアウト値をロードしておく、タイムアウト期間の経過時に割り込みが生成され、ドライバーに通知されます。

性能データの提供およびデバッグ支援を目的として、HDCP 2.2 トランスミッターおよびレシーバーは HDCP 2.2 Timer カウンターに基づいてドライバーのログ バッファにタイムスタンプを挿入します。ログ バッファに書き込まれるすべてのイベントに対して相対時間が記録されます。ログを表示すると、この相対時間および算出された差分時間が出力されます。

AXI4-Lite Crossbar

これは AXI Crossbar LogiCORE IP です。これにより、HDCP 2.2 から外部に 1 つの AXI4-Lite スレーブ インターフェイスが提供されるため、プロセッサは連続したアドレス空間で内部スレーブのステータス/制御レジスタにアクセスできます。

規格

ザイリンクスの HDCP 2.2 は、Digital Content Protection (DCP) LLC が発行した HDCP 2.2 仕様『High-bandwidth Digital Content Protection, Mapping HDCP to HDMI, Revision 2.2』[参照 12] に準拠しています。

また、ザイリンクスの HDCP 2.2 は DCP LLC が変更点とテスト ベクターを文書化して 2015 年 2 月 9 日に発行した『Errata to HDCP 2.2 on HDMI Specification, Version 2』[参照 13] にも準拠しています。

性能とリソース使用状況

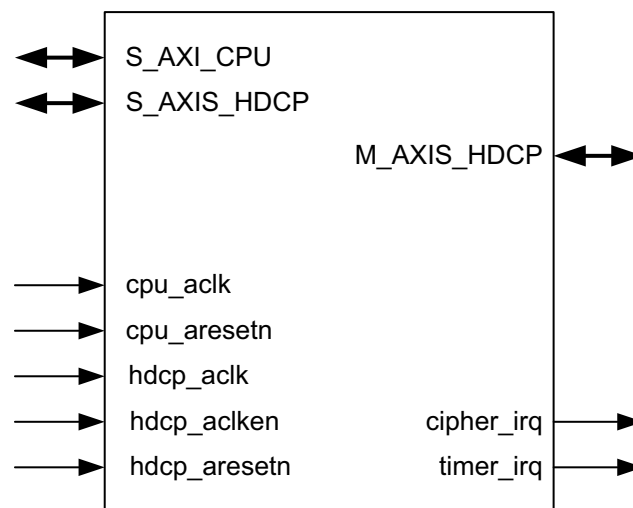
性能およびリソース使用状況の詳細は、次のウェブ ページを参照してください。

- [Performance and Resource Utilization \(トランスミッター\)](#)
- [Performance and Resource Utilization \(レシーバー\)](#)

ポートの説明

図 2-3 に、HDCP 2.2 トランスミッターおよびレシーバーのポート図を示します。制御/ステータスおよびデータフロー用として次の 3 つのインターフェイスを使用します。

- 制御/ステータス用の AXI4-Lite スレーブ インターフェイス (S_AXI_CPU)
- データ入力用の AXI4-Stream スレーブ インターフェイス (S_AXIS_HDCP)
- データ出力用の AXI4-Stream マスター インターフェイス (M_AXIS_HDCP)



X16256-091616

図 2-3: HDCP のピン配置

表 2-1 に、クロッキング、リセット、および割り込みに使用するシステム ポートを示します。AXI4-Lite ホスト インターフェイスと割り込み信号は `cpu_aclk` ドメインに同期します。AXI4-Stream Ingress/Egress インターフェイスは `hdcp_aclk` ドメインに同期します。

表 2-1: HDCP 2.2 ポート

名称	方向	説明
<code>cpu_aclk</code>	入力	AXI4-Lite クロック
<code>cpu_aresetn</code>	入力	AXI4-Lite リセット (アクティブ Low)
<code>cipher_irq</code>	出力	Cipher 割り込み (<code>cpu_aclk</code> に同期)。この割り込みは現在トランスミッターとレシーバーのどちらも使用しておらず、常に Low に駆動されます。
<code>timer_irq</code>	出力	Timer 割り込み (<code>cpu_aclk</code> に同期)。この割り込みは現在トランスミッターでウォッチドッグ タイマーとしてのみ使用されており、レシーバーでは常に Low に駆動されます。
<code>hdcp_aclk</code>	入力	AXI4-Stream クロック
<code>hdcp_aclken</code>	入力	AXI4-Stream クロック イネーブル
<code>hdcp_aresetn</code>	入力	AXI4-Stream リセット (アクティブ Low)

表 2-2 に、制御およびステータスに使用するホスト インターフェイスの AXI4-Lite ポートを示します。

表 2-2: HDCP 2.2 ホスト インターフェイス

名称	方向	説明
<code>s_axi_cpu_awaddr[17:0]</code>	入力	AXI4-Lite 書き込みアドレス
<code>s_axi_cpu_awvalid</code>	入力	AXI4-Lite 書き込みアドレス Valid 信号
<code>s_axi_cpu_awready</code>	出力	AXI4-Lite 書き込みアドレス Ready 信号
<code>s_axi_cpu_wdata[31:0]</code>	入力	AXI4-Lite 書き込みデータ
<code>s_axi_cpu_wstrb[3:0]</code>	入力	AXI4-Lite 書き込みストローブ
<code>s_axi_cpu_wvalid</code>	入力	AXI4-Lite 書き込み Valid 信号
<code>s_axi_cpu_wready</code>	出力	AXI4-Lite 書き込み Ready 信号
<code>s_axi_cpu_bresp[1:0]</code>	出力	AXI4-Lite 書き込み応答
<code>s_axi_cpu_bvalid</code>	出力	AXI4-Lite 書き込み応答 Valid 信号
<code>s_axi_cpu_bready</code>	入力	AXI4-Lite 書き込み応答 Ready 信号
<code>s_axi_cpu_araddr[17:0]</code>	入力	AXI4-Lite 読み出しアドレス
<code>s_axi_cpu_arvalid</code>	入力	AXI4-Lite 読み出しアドレス Valid 信号
<code>s_axi_cpu_arready</code>	出力	AXI4-Lite 読み出しアドレス Ready 信号
<code>s_axi_cpu_rdata[31:0]</code>	出力	AXI4-Lite 読み出しデータ
<code>s_axi_cpu_rresp[1:0]</code>	出力	AXI4-Lite 読み出し応答
<code>s_axi_cpu_rvalid</code>	出力	AXI4-Lite 読み出し Valid 信号
<code>s_axi_cpu_rready</code>	入力	AXI4-Lite 読み出し Ready 信号

表 2-3 に、HDCP Egress インターフェイスの AXI4-Stream ポートを示します。HDCP 2.2 トランスミッターでは、暗号化されていないデータが Egress インターフェイス経由で HDCP 2.2 トランスミッターに入力されます。反対に、HDCP 2.2 レシーバーでは暗号化されたデータが HDCP Egress インターフェイスに入力されます。

表 2-3: HDCP 2.2 Egress インターフェイス

名称	方向	説明
s_axis_hdcp_tdata[95:0]	入力	AXI4-Stream Egress データ
s_axis_hdcp_tid	入力	AXI4-Stream Egress ID
s_axis_hdcp_tlast	入力	AXI4-Stream Egress Last 信号
s_axis_hdcp_tready	出力	AXI4-Stream Egress Ready 信号
s_axis_hdcp_tstrb[3:0]	入力	AXI4-Stream Egress ストロープ
s_axis_hdcp_tuser[31:0]	入力	AXI4-Stream Egress ユーザー ビット [31:28] = キープアウト領域インジケータ (デバッグ専用) ビット [27:24] = オポチュニティ ウィンドウ インジケータ ビット [23:20] = EESS 制御信号 (レーン 3) ビット [19] = 垂直同期 (レーン 3) ビット [18] = 水平同期 (レーン 3) ビット [17:14] = EESS 制御信号 (レーン 2) ビット [13] = 垂直同期 (レーン 2) ビット [12] = 水平同期 (レーン 2) ビット [11:8] = EESS 制御信号 (レーン 1) ビット [7] = 垂直同期 (レーン 1) ビット [6] = 水平同期 (レーン 1) ビット [5:2] = EESS 制御信号 (レーン 0) ビット [1] = 垂直同期 (レーン 0) ビット [0] = 水平同期 (レーン 0)
s_axis_hdcp_tvalid	入力	AXI4-Stream Egress Valid 信号

表 2-4 に、Ingress インターフェイスの AXI4-Stream ポートを示します。HDCP 2.2 トランスミッターでは、暗号化されたデータが Ingress インターフェイスから出力され、リンク上に送信されます。反対に、HDCP 2.2 レシーバーでは復号化されたデータが HDCP Ingress インターフェイスから出力され、ダウンストリームデバイスへ送信されます。

表 2-4: HDCP 2.2 Ingress インターフェイス

名称	方向	説明
m_axis_hdcp_tdata[95:0]	出力	AXI4-Stream Ingress データ
m_axis_hdcp_tid	出力	AXI4-Stream Ingress ID
m_axis_hdcp_tlast	出力	AXI4-Stream Ingress Last 信号
m_axis_hdcp_tready	入力	AXI4-Stream Ingress Ready 信号
m_axis_hdcp_tstrb[3:0]	出力	AXI4-Stream Ingress ストロープ
m_axis_hdcp_tuser[31:0]	出力	AXI4-Stream Ingress ユーザー ビット フィールドはポート s_axis_hdcp_tuser と同じ。
m_axis_hdcp_tvalid	出力	AXI4-Stream Ingress Valid 信号

IP を使用するデザイン

この章では、IP を使用してデザインを完成させるためのガイドラインおよびその他の情報を紹介します。

HDCP 2.2 はサブシステムの一部として提供されているため、HDCP 2.2 に直接アクセスするためのインターフェイスは存在せず、HDMI Transmitter/Receiver Subsystem をコンフィギュレーションすると HDCP 2.2 の動作が自動的に設定されます。HDMI Transmitter/Receiver Subsystem を使用してシステムを構築する方法の詳細は、HDMI 製品ガイド (『HDMI 1.4/2.0 Transmitter Subsystem v2.0 製品ガイド』(PG235) [参照 1] および『HDMI 1.4/2.0 Receiver Subsystem v2.0 製品ガイド』(PG236) [参照 2]) を参照してください。

HDCP 2.2 トランスミッターおよびレシーバーは、Digital Content Protection (DCP) LLC が発行するプロダクションキーを使用してプログラムする必要があります。次に、ユーザー アプリケーションを使用してロードする必要のあるキーの一覧を示します。

- HDCP 2.2 トランスミッター
 - 16 バイト: グローバル定数 (lc_{128})
- HDCP 2.2 レシーバー
 - 16 バイト (トランスミッターと同じ): グローバル定数 (lc_{128})
 - 320 バイト: RSA 秘密キー ($kpriv_{rx}$)
 - 522 バイト: RSA 公開キー証明書 ($cert_{rx}$)

HDCP 2.2 ドライバーにロードするキーを安全に格納および取り出すためのインフラストラクチャは、ユーザー アプリケーションで用意する必要があります。HDCP 2.2 トランスミッターおよびレシーバーには、キーの格納と取り出しに関するメカニズムはありません。キーの機密性と完全性はユーザー アプリケーションで維持する必要があります。キーをソフトウェアドライバーにロードする方法の詳細は、付録 A 「アプリケーション ソフトウェア開発」を参照してください。

キーを安全に格納および取り出すリファレンス デザインの使用の詳細は、アプリケーション ノート『Kintex UltraScale FPGA GTH トランシーバーを使用した HDMI 2.0 の実装』(XAPP1275) [参照 3] を参照してください。

一般的なデザインガイドライン

HDCP 2.2 は HDMI サブシステムの一部に含まれるため、HDCP 2.2 IP はこのサブシステムによってハードウェアに統合されます。HDCP 2.2 IP のインスタンスシートおよび HDMI IP との接続は、HDMI サブシステムで行われます。ハードウェアはサブシステムとして抽象化されるため、HDCP 2.2 の統合はソフトウェアドライバ API を使用してプロダクション キーをロードし、ステート マシン実行中に実行されるコールバックをバインドすることが主となります。このセクションでは、HDCP 2.2 ステート マシンを初期化および実行する上で理解しておくべき重要なソフトウェア統合コンセプトについて説明します。

トランスミッター キー

HDCP 2.2 トランスミッターの Cipher にはグローバル定数 (lc_{128}) をロードする必要があります。すべての HDCP 2.2 デバイスは、AES-128 Cipher で暗号化を実行するためのパラメーターとしてこのグローバル定数を使用します。これは HDCP Adopter に対してのみ提供される 128 ビットの値で、バイト オーダーはビッグ エンディアンフォーマットです。このキーの機密性と完全性はユーザー アプリケーションで維持する必要があります。HDCP 2.2 ドライバには、ドライバ初期化プロセス中にこのキーをロードする機能があります。キーは HDCP 2.2 トランスミッターのステート マシン実行開始前にロードする必要があります。キーを正しい手順でロードしなかった場合や誤ったキーをロードした場合、認証とキー交換のプロセスは正しく完了しますがトランスミッターの Cipher で正しい暗号ワードが生成されません。

レシーバー キー

HDCP 2.2 レシーバーの Cipher にも、前のセクションで述べたのと同じグローバル定数 lc_{128} をロードする必要があります。このキー以外にも、すべての HDCP 2.2 レシーバー デバイスにデバイス秘密キー ($k_{priv_{rx}}$) とデバイス公開キー証明書 ($cert_{rx}$) の 2 つをロードする必要があります。

計算時間を短縮するため、DCP から提供されるデバイス秘密キーは中国剰余定理 (CRT) を使用したべき剰余の実行に適したフォーマットを採用しています。秘密キーは HDCP Adopter に対してのみ提供される 320 バイトの値で、 p 、 q 、 dP 、 dQ 、 $qInv$ の 5 つの値で構成されます。バイト オーダーはビッグ エンディアンフォーマットです。このキーの機密性と完全性はユーザー アプリケーションで維持する必要があります。HDCP 2.2 ドライバには、ドライバ初期化プロセス中にこのキーをロードする機能があります。キーは HDCP 2.2 トランスミッターのステート マシン実行開始前にロードする必要があります。キーを正しい手順でロードしなかった場合や誤ったキーをロードした場合、認証およびキー交換プロセス (具体的には AKE_No_Stored_km メッセージによる RSA 暗号化されたマスター キー (k_m) の交換) に失敗します。

デバイス公開キー証明書には一意のレシーバー ID、レシーバー公開キー、および証明書の検証に使用する DCP シグネチャが含まれます。公開キー証明書は 522 バイトの値で、バイト オーダーはビッグ エンディアンフォーマットです。公開キー証明書の完全性はユーザー アプリケーションで維持する必要がありますが、機密性は必ずしも必要ではありません。HDCP 2.2 ドライバには、ドライバ初期化プロセス中にこのキーをロードする機能があります。キーは HDCP 2.2 トランスミッターのステート マシン実行開始前にロードする必要があります。キーを正しい手順でロードしなかった場合や誤ったキーをロードした場合、認証およびキー交換プロセス (具体的には AKE_Send_Cert メッセージによる公開キー証明書の交換) に失敗します (トランスミッターによる署名検証に失敗するため)。

SRM (System Renewability Message)

HDCP 2.2 トランスミッターには不正機器を排除するチェック動作が必須です。これは、ダウンストリーム レシーバーが DCP によって排除された機器のリスト (ブラックリスト) に登録されている場合、認証を強制終了するものです。SRM には、DCP によって不正機器と確認されたレシーバー ID のリストが含まれており、HDCP 2.2 トランスミッターはこれに該当するレシーバーを排除する必要があります。SRM に記録されるレシーバー ID の数は SRM のヘッダーで定義します。SRM はユーザー アプリケーションで管理して完全性を維持する必要があります。HDCP 2.2 ドライバには、ドライバ初期化プロセス中に完全な SRM をロードする機能があります。SRM は、HDCP 2.2 トランスミッターのステート マシン実行開始前にロードする必要があります。SRM を正しい手順でロードしなかった場合や誤った SRM をロードした場合、あるいはレシーバー ID がブラックリストに登録されている場合、認証プロトコルは失敗し、トランスミッターによって中止されます。

ステート マシンの性能

HDCP 2.2 ステート マシンは、ユーザー アプリケーションが定期的にポーリング関数を呼び出して実行します。ステート マシンの性能は、ポーリング関数を呼び出す頻度によって決まります。いずれかのインターフェイスに対してポーリング関数の呼び出しが遅れると性能が低下するだけでなく、仕様で定義されたタイミング要件に違反して認証エラーとなることもあるため、注意が必要です。HDCP 2.2 IP のトランスミッター/レシーバー ドライバーにはログ バッファがあり、これを使用してプロトコルの動作と性能を評価できます。

HDCP 2.2 仕様では、認証実行中のメッセージ交換に関する制限時間が定義されており、トランスミッターまたはレシーバーはこれに従って動作します。プロトコルで定義されている最も厳格な制限時間は LC_Send_L_prime メッセージの交換で、トランスミッターは LC_Init メッセージの最後のバイトを送信してから 20ms 以内にこのメッセージを受信する必要があります。したがって、認証実行中にステート マシンを正しく動作させるには、ユーザー アプリケーションで少なくとも 20ms ごとに 1 回は HDCP 2.2 ポーリング関数を呼び出す必要があります。HDCP 2.2 アップストリームまたはダウンストリーム インターフェイスが複数あるアプリケーションでは、どのインターフェイスに対しても十分な頻度でポーリング関数が実行されるようにスケジュールする必要があります。

トランスミッターとレシーバーの動作

最上位の HDCP 2.2 トランスミッターは、次のクラスのデバイスとの間で認証が可能です。

- HDCP 2.2 エンドポイント レシーバー
- HDCP 2.2 リピーター アップストリーム インターフェイス

認証ステート マシンの動作は、トランスミッターが接続するデバイスのクラスに応じて多少異なります。リピーターとの認証ではトポロジ情報およびストリーム管理情報の交換が必要ですが、エンドポイント レシーバーとの認証ではこれらは必要ありません。HDCP 2.2 トランスミッター ドライバーでは、コールバック関数を登録してステート マシンの各種遷移時に実行できます。表 3-1 に、トランスミッターのコールバック イベントをまとめます。詳細は、付録 A 「アプリケーション ソフトウェア開発」を参照してください。

表 3-1: トランスミッターのコールバック イベント

コールバック イベント	説明
Authenticated	ステート マシンが Authenticated ステートに遷移すると実行されます。
Unauthenticated	ステート マシンがいずれかのステートから Unauthenticated ステートに戻ると実行されます。
Topology Available	アップストリームへの伝搬が必要なトポロジ情報が存在する場合に実行されます。

高価値コンテンツの送信をいつ開始するかは、最上位の HDCP 2.2 トランスミッターが決定します。トランスミッターはステート マシンが Authenticated ステートに遷移するまで暗号化を有効にできません。したがって、ステート マシンが Authenticated ステートに遷移するまでは低価値コンテンツしかダウンストリームに送信できません。トランスミッターが高価値コンテンツの送信をいつ開始するかは、Authenticated コールバックに基づいて判断します。低価値コンテンツは暗号化しなくても送信できますが、高価値コンテンツは常に暗号化して送信する必要があります。反対に、トランスミッターが高価値コンテンツの送信をいつ停止するかは、Unauthenticated コールバックに基づいて判断します。

HDCP 2.2 エンドポイント レシーバーは、次のクラスのデバイスとの間で認証が可能です。

- 最上位の HDCP 2.2 トランスミッター
- HDCP 2.2 リピーター ダウンストリーム インターフェイス

トランスミッターとは異なり、レシーバーのステート マシンは最上位トランスミッターであれリピーター インターフェイスであれ同じ方法で認証を実行します。表 3-2 に、HDCP 2.2 レシーバー ドライバーのコールバック イベントをまとめます。詳細は、付録 A 「アプリケーション ソフトウェア開発」を参照してください。

注記: 一部のコールバックは、レシーバーがリピーター アップストリーム インターフェイスとして動作している場合のみ利用可能です。

表 3-2: レシーバーのコールバック イベント

コールバック イベント	説明
Authenticated	ステート マシンが Authenticated ステートに遷移すると実行されます。
Unauthenticated	ステート マシンが Unauthenticated ステートに遷移すると実行されます。
Authentication Request	アップストリーム トランスミッターから認証要求 (AKE_Init) を受信すると実行されます。
Encryption Update	レシーバーが Authenticated ステートのときに暗号化ステータスが Unencrypted から Encrypted (または Encrypted から Unencrypted) へ変化すると実行されます。1 秒タイマー割り込みに基づきます。
Topology Update	ステート マシンがアップストリームへ伝搬するトポロジ情報を待機しているときに実行されます。リピーター モードの場合のみ利用できます。
Stream Management Request	コンテンツのストリーム管理情報メッセージ (RepeaterAuth_Stream_Manage) を受信すると実行されます。リピーター モードの場合のみ利用できます。

HDCP 2.2 レシーバーは Authenticated ステートに遷移した時点から受信コンテンツの復号化が可能です。いつコンテンツの暗号化を開始するかはトランスミッターが決定するため、ステート マシンが Authenticated ステートに遷移した後のレシーバーはいつでも復号化を実行できるようにしておく必要があります。これは HDCP 2.2 IP によって自動的に行われます。図 3-1 に示すように、リピーター デザインの一部としてレシーバーを非リピーター モードで使用する場合、アップストリーム インターフェイスとダウンストリーム インターフェイスが分離されていることに注意が必要です。この場合、アップストリーム インターフェイスとダウンストリーム インターフェイスの認証プロトコルは独立しており、最上位トランスミッター (Tx1) からダウンストリーム レシーバー (Rx1) の認証ステータスはわかりません。

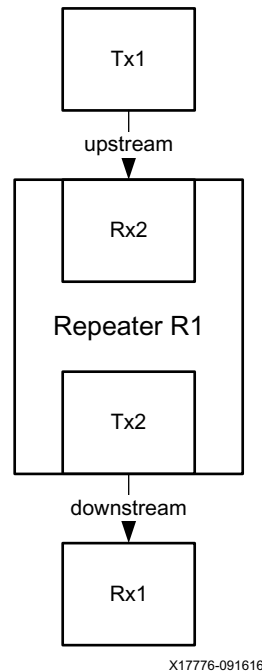


図 3-1: ダウンストリーム インターフェイスが 1 つの分離リピーター トポロジ

分離リピータートポロジでは、ダウンストリーム インターフェイスで認証が完了していない場合、ダウンストリーム インターフェイスのトランスミッター (Tx2) が保護されたコンテンツの送信をブロックする必要があります。この場合、ブルー スクリーンなどの低価値コンテンツをダウンストリーム インターフェイスに送信します。HDCP 2.2 トランスミッターの Cipher にはブランク出力を生成してコンテンツをブロックする機能があります。ダウンストリーム インターフェイスでコンテンツをブロックするかどうかは、アップストリーム インターフェイスの暗号化ステータスに基づき、レシーバーの **Authenticated** および **Unauthenticated** コールバック イベントを使用して決定します。ステート マシンが **Authenticated** ステートに遷移すると、レシーバー ドライバーは 1 秒周期タイマー割り込みをセットします。このタイマー割り込みを使用して受信フレームの暗号化ステータスをチェックします。この期間中に暗号化ステータスが **Unencrypted** から **Encrypted**、または **Encrypted** から **Unencrypted** へ変化すると、**Encryption Update** コールバック イベントが実行されます。**Encryption Update** コールバックにおいて、アップストリーム インターフェイスが認証および暗号化されており、ダウンストリーム インターフェイスが認証されていない場合、コンテンツをブロックする必要があります。

分離リピータートポロジでは、ダウンストリーム インターフェイスでいつ暗号化を有効にするかを検討することも重要です。暗号化を有効にする方法には次の 2 つのオプションがあります。

- オプション 1: ダウンストリーム インターフェイスが **Authenticated** ステートに遷移した直後に暗号化を有効にする。
- オプション 2: ダウンストリーム インターフェイスが **Authenticated** ステートに遷移し、アップストリーム インターフェイスが暗号化されたフレームの受信を開始してから暗号化を有効にする。

オプション 1 は保護されたコンテンツを暗号化しないで送信してしまうのを防ぐことができるため、この方法を推奨します。アップストリーム レシーバー (Rx2) は、**Encryption Update** コールバック イベントに基づいて 1 秒間隔で受信フレームの暗号化を検出します。したがって、1 秒間隔の途中で暗号化されたフレームを受信した場合、レシーバーは検出できません。また、悪意のあるソースが 1 フレームおきなど何らかのパターンで暗号化して、レシーバーが受信フレームの暗号化を検出できないように細工をする可能性もあります。



推奨: 上記の理由により、分離リピーターではなく本来のリピーター構成を使用することを推奨します。

次のセクションでは、リピーター モードでのトランスミッターとレシーバーの動作について説明します。

リピーターとコンバーターの動作

HDCP 2.2 リピーターには 1 つのアップストリーム インターフェイスと 1 つまたは複数のダウンストリーム インターフェイスがあります (図 3-2)。リピーターはカスケード接続が可能です (最大で **DEPTH=4**、**DEVICE_COUNT=31** まで)。HDCP 2.2 IP のトランスミッターとレシーバーは、動作中にユーザー アプリケーションによってリピーターモードに設定できます。リピーターモードを有効にすると、認証ステート マシンの動作がトランスミッターの場合はリピーター ダウンストリーム インターフェイスとしての動作へと変化し、レシーバーの場合はリピーター アップストリーム インターフェイスとしての動作へと変化します。

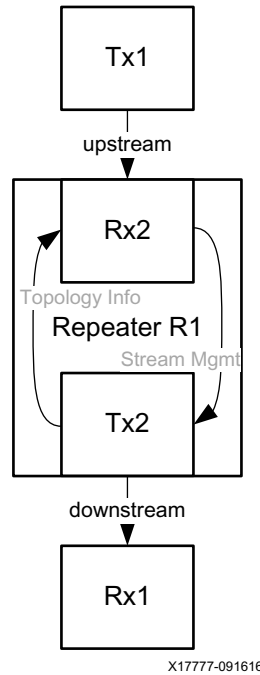
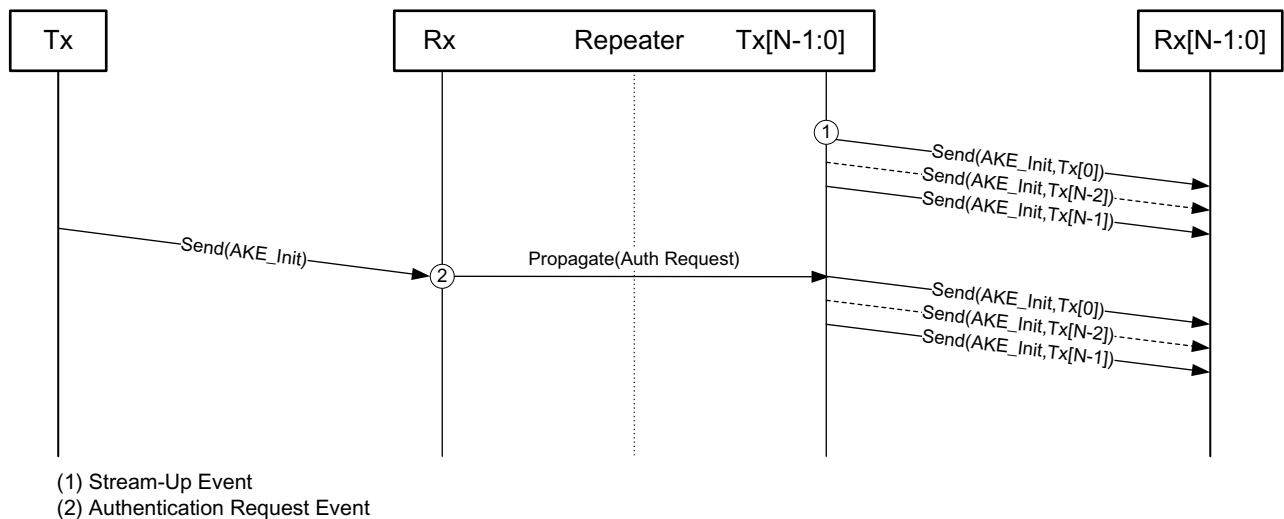


図 3-2: ダウンストリーム インターフェイスが1つのリピーター

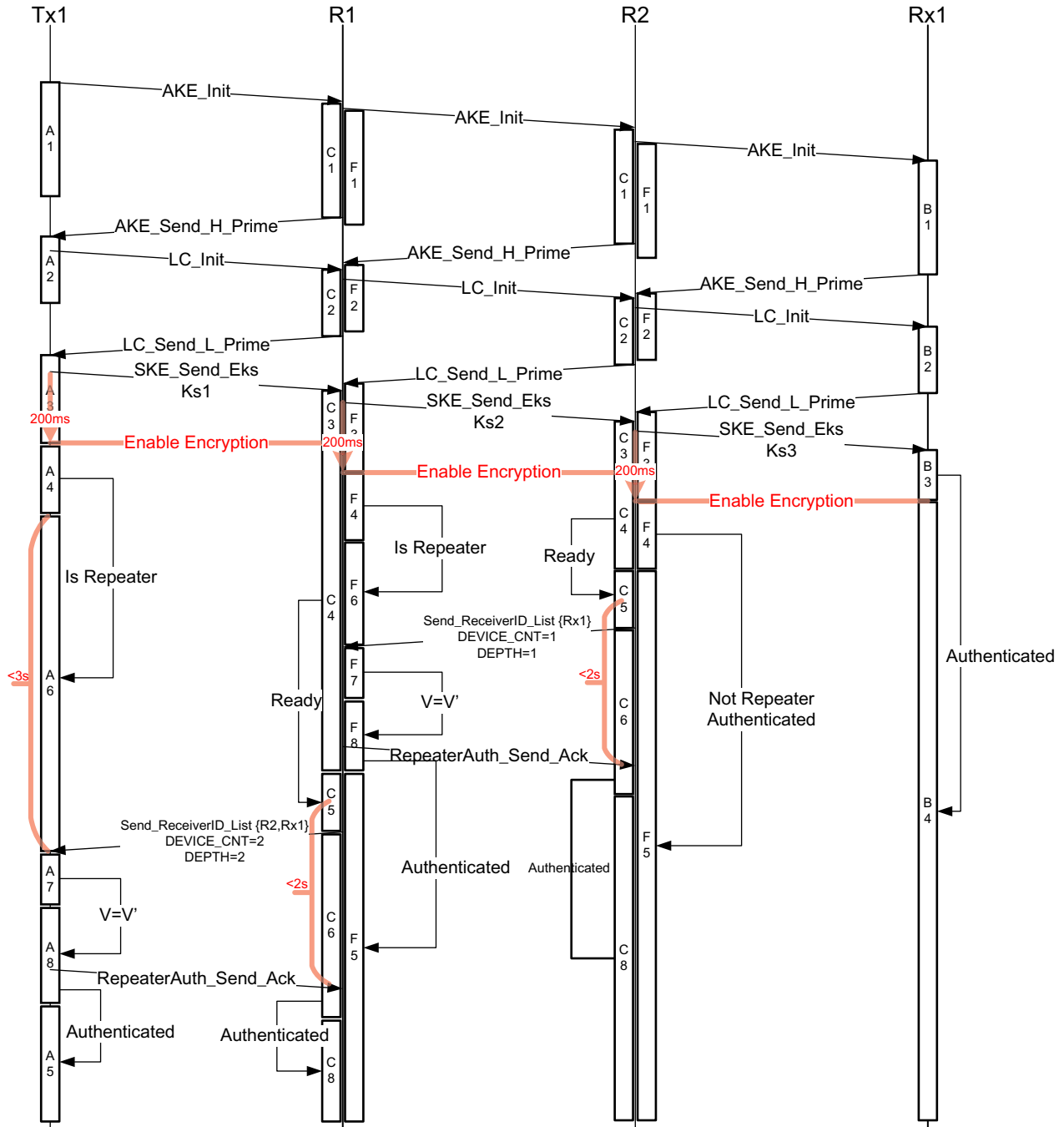
リピーター モードでも前のセクションで定義したコールバック イベントと同じものを使用します。図 3-3 に、ユーザー アプリケーションがダウンストリーム認証要求をトリガーする方法を示します。1つ目の方法は、ダウンストリーム インターフェイスがコンテンツを送信可能な状態になると (Stream-Up イベント)、ただちに認証をトリガーする方法です。Stream-Up イベントは HDMI トランスミッターによって生成されます。2つ目の方法は、HDCP 2.2 の Authentication Request イベント コールバックを使用して認証要求を伝搬するというものです。この方法では、まだ Authenticated ステートになっていないダウンストリーム インターフェイスに対してのみ認証をトリガーする必要があります。



X17778-091616

図 3-3: リピーター ダウンストリーム認証要求

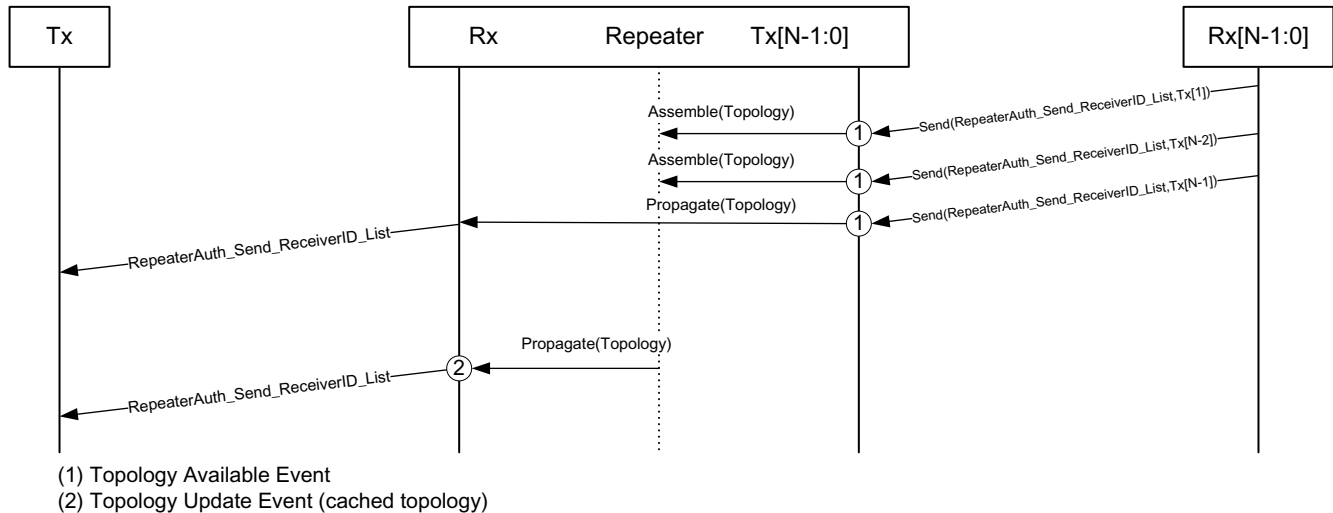
図 3-4 に、2つのリピーター (R1、R2) と1つのエンドポイントレシーバー (Rx1) で構成したトポロジにおけるアップストリームトポロジ伝搬を示します。リピーターのダウンストリームステートマシンはトランスミッターのステートマシンとほぼ同じですが、不正機器の排除チェックが不要な点が異なります。これに対し、リピーターのアップストリームステートマシンにはトポロジ情報をアップストリームに伝搬するためのステートとストリーム管理情報をダウンストリームに伝搬するためのステートが存在します。



X17779-091616

図 3-4: リピーターのアップストリームトポロジ伝搬 (DEPTH=2、DEVICE_COUNT=2)

トポロジ情報はリピータートポロジの最下位から最上位トランスミッターへとアップストリームに伝搬されます。ダウンストリーム インターフェイスが認証されてトポロジ情報が検証されると、トポロジ内の各リピーターがトポロジ情報をアップストリームに送信します。HDCP 2.2 トランスミッタードライバーには、各ダウンストリーム インターフェイスのトポロジ情報を収集する手段としてコールバック イベントの **Topology Available** があります (図 3-5)。トポロジ伝搬では、最上位トランスミッターがトポロジ情報を受信した時点ですべてのダウンストリーム インターフェイスの認証が完了して暗号化を実行できる状態であり、それぞれのトポロジ情報の検証が完了していることとなります。トポロジ内のすべてのダウンストリーム デバイスの認証が完了していることから、最上位トランスミッターは保護されたコンテンツの送信を安全に開始できることが保証されます。対照的に、前のセクションで説明した分離リピーターの場合、最上位トランスミッターからダウンストリーム デバイスの認証ステータスはわかりません。



X17780-091616

図 3-5: リピーターのアップストリーム トポロジ伝搬

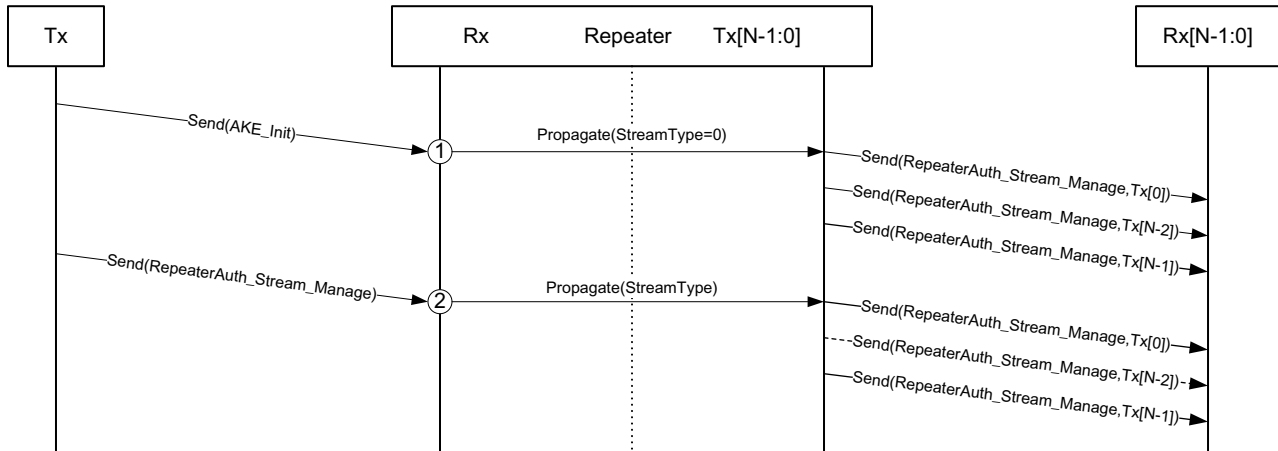
HDCP 2.2 プロトコルでは、リピーターがダウンストリーム インターフェイスのトポロジ情報をキャッシュすることが認められています。HDCP 2.2 トランスミッタードライバーは、リピーター認証中に受信した最新のトポロジ情報を自動的にキャッシュします。キャッシュしたトポロジ情報を用いることで、認証済みダウンストリーム インターフェイスの再認証が不要になります。このため、トポロジ変更に伴うトポロジ情報のアップストリーム伝搬が高速化されます。トポロジ情報のキャッシュによる高速化のメリットは、ユーザー アプリケーションにももたらされます。図 3-5 には、リピーターのアップストリーム インターフェイスで実行された **Topology Update** コールバック イベントを使用してキャッシュ済みのトポロジ情報を伝搬する方法も示しています。

トポロジ情報メッセージには、トポロジに接続されているデバイスのタイプをソースに通知するためのフラグが含まれます。これらのフラグは、次のトポロジ情報を示します。

- HDCP 1.x デバイスがダウンストリームに接続されているかどうか
- HDCP 2.0 リピーターがダウンストリームに接続されているかどうか
- トポロジの最大数を超えているかどうか

HDCP 2.2 で保護されたコンテンツを送信するかどうかは、これらのトポロジフラグに基づいて最上位トランスミッターが決定します。たとえば HDCP 1.x デバイスがダウンストリームに接続されていることがトポロジ情報で示されている場合、最上位トランスミッターは保護されたコンテンツを送信しないように決定できます。

ストリーム管理情報は最上位トランスミッターからダウンストリームへ伝搬されます。最上位トランスミッターはコンテンツが Type 0 と Type 1 のどちらであるかを判断する必要があります。Type 0 コンテンツはすべての HDCP デバイスに送信できますが、Type 1 コンテンツは HDCP 2.2 デバイスにしか送信できません。HDCP 2.2 レシーバードライバはストリーム管理要求を受信すると (図 3-6 の 2 番目のイベント)、コールバック イベント Stream Management Request を実行します。このコールバックを実行すると、ユーザーアプリケーションはストリームタイプを抽出してダウンストリーム インターフェイスに対してストリームをブロックするかどうかを判断できます。HDCP 1.4 から HDCP 2.2 への変換の場合、トランスミッターはストリーム管理情報を送信しません。したがって、Authentication Request イベントの後にすべてのダウンストリーム インターフェイスに対してストリームタイプが 0 に設定されます。



- (1) Authentication Request Event (when upstream is HDCP 1.x)
- (2) Stream Management Request Event (when upstream is HDCP 2.2)

X17781-091616

図 3-6: リピーターのダウンストリームコンテンツストリーム管理の伝搬

HDCP 2.2 プロトコルは HDCP 2.2 と HDCP 1.x の間で双方向の変換をサポートします。HDMI Subsystem のカスタマイズでは、HDCP 1.x と HDCP 2.2 を共に有効にするなど、任意の HDCP プロトコルをサポートできます。リピーターのダウンストリーム インターフェイスに接続されたデバイスに対しては、サポートされるプロトコルに基づいてコンテンツを適宜ブロックする必要があります。表 3-3 に、リピーターとレシーバーでサポートされるプロトコルの組み合わせごとにダウンストリーム インターフェイスでコンテンツがブロックされるかどうかを示します。リピーターがサポートするプロトコルをレシーバーが 1 つもサポートしていない場合、ダウンストリーム インターフェイスがブロックされます。

注記: HDCP 2.2 IP は HDCP 1.x プロトコルをサポートしません。プロトコル変換をサポートするには、HDCP 1.x もサブシステムに含める必要があります。

表 3-3: リピーターのダウンストリームでコンテンツがブロックされるプロトコルの組み合わせ

リピーターがサポートするプロトコル	レシーバーがサポートするプロトコル	ブロック
HDCP 2.2	HDCP 2.2	しない
HDCP 2.2	HDCP 1.x	する
HDCP 2.2	なし	する
HDCP 1.x	HDCP 1.x	しない
HDCP 1.x	HDCP 2.2	する
HDCP 1.x	なし	する
HDCP 1.x および HDCP 2.2	HDCP 1.x	しない
HDCP 1.x および HDCP 2.2	HDCP 2.2	しない
HDCP 1.x および HDCP 2.2	HDCP 1.x および HDCP 2.2	しない
HDCP 1.4/2.2	なし	する

リピーターのダウンストリーム インターフェイスに HDCP 対応デバイスが接続されていない場合、リピーターは次のいずれかを実行できます。

- オプション 1: リピーターのアップストリーム インターフェイスで HPD をディアサートする。
- オプション 2: アップストリーム インターフェイスでリピーター サポートを無効にし、保護されたコンテンツをダウンストリームに送信しない。

オプション 1 は最もシンプルな方法ですが、保護されていないコンテンツも含め、ダウンストリームにはコンテンツが一切送信されません。HDCP 対応デバイスを接続すると HPD がアサートされ、リピーター認証が実行されます。オプション 2 は保護されていないコンテンツをダウンストリームに送信できる利点がありますが、アップストリーム インターフェイスでコンテンツ保護が有効にされた場合にコンテンツをブロックする手間がかかります。HDCP 対応デバイスを接続すると、アップストリーム インターフェイスはリピーター モードに戻り、少なくとも 100ms の間 HPD をトグルしてアップストリーム トランスミッターに対して再認証が必要なことを通知します。アプリケーションでどちらのオプションを採用するかは、要件に応じてユーザーが決定する必要があります。

自動プロトコル切り替え

HDMI Receiver Subsystem には HDCP プロトコルを自動的に切り替える機能があります。HDMI レシーバーは、トランスミッターがアクセスを試みているアドレス範囲に基づいてアクティブなプロトコルを変更します。HDCP 1.4 が占有する I2C アドレス範囲は 0x00 ~ 0x44 で、HDCP 2.2 が占有するアドレス範囲は 0x50 ~ 0x81 です。HDCP 2.2 トランスミッターは、認証を開始する前に HDCP 2 バージョンレジスタを正しく読み出すことがプロトコルで規定されています。HDMI レシーバーは、このレジスタの最初の読み出しを検出すると自動的に HDCP 2.2 をアクティブなプロトコルとして設定します。同様に、HDCP 1.x トランスミッターは認証を開始する前に HDCP 1.x レジスタを正しく読み出す必要があります。HDCP 1.x レジスタの読み出しに成功すると、HDMI レシーバーは自動的に HDCP 1.x をアクティブなプロトコルとして設定します。

注記: HDCP 2.2 IP は HDCP 1.x プロトコルをサポートしません。自動プロトコル切り替えをサポートするには、HDCP 1.x もサブシステムに含める必要があります。

クロッキング

HDCP 2.2 IP には次の 2 つのクロック ドメインがあります。

- `cpu_aclk`: ホスト プロセッサのクロック ドメインで、制御およびステータスに使用します。通常、このクロック ドメインの周波数は固定です。HDCP 2.2 の IP コアのうち、Random Number Generator、Montgomery Modular Multiplier、および Timer はこのクロック ドメインでのみ動作します。
- `hdcp_aclk`: オーディオ/ビジュアル データのクロック ドメインで、Cipher データパスのクロックとして使用します。このクロック ドメインの周波数は HDMI リンクのクロック周波数によって変化します。

リセット

HDCP 2.2 ブロックには 2 つのリセットがあります。

- `cpu_aresetn`: アクティブ Low のリセット入力で、`cpu_aclk` ドメインに同期します。
- `hdcp_aresetn`: アクティブ Low のリセット入力で、`hdcp_aclk` ドメインに同期します。

デザイン フローの手順

この章では、IP のカスタマイズと生成、制約、およびシミュレーション/合成/インプリメンテーションの手順について説明します。一般的な Vivado® デザイン フローおよび IP インテグレーターの詳細は、次の Vivado Design Suite ユーザー ガイドを参照してください。

- 『Vivado Design Suite ユーザー ガイド: IP インテグレーターを使用した IP サブシステムの設計』(UG994) [参照 4]
- 『Vivado Design Suite ユーザー ガイド: IP を使用した設計』(UG896) [参照 5]
- 『Vivado Design Suite ユーザー ガイド: 入門』(UG910) [参照 6]
- 『Vivado Design Suite ユーザー ガイド: ロジック シミュレーション』(UG900) [参照 7]

IP のカスタマイズおよび生成

ここでは、ザイリンクス ツールを使用し、Vivado Design Suite で IP をカスタマイズおよび生成する方法について説明します。

Vivado IP インテグレーターで IP をカスタマイズおよび生成する場合は、『Vivado Design Suite ユーザー ガイド: IP インテグレーターを使用した IP サブシステムの設計』(UG994) [参照 4] を参照してください。IP インテグレーターは、デザインの検証または生成時に一部のコンフィギュレーション値を自動的に計算する場合があります。値が変わるかどうかなを確認するには、この章のパラメーターの説明を参照してください。パラメーター値を確認するには、Tcl コンソールから `validate_bd_design` コマンドを実行してください。

IP はユーザー デザインに合わせてカスタマイズできます。それには、IP に関連する各種パラメーターの値を次の手順に従って指定します。

1. Vivado IP カタログで [HDMI Transmitter Subsystem] または [HDMI Receiver Subsystem] を選択します。HDCP 2.2 IP は HDMI Subsystem の一部として自動的に生成されます。
2. 選択した IP をダブルクリックするか、ツールバーまたは右クリック メニューから [Customize IP] コマンドをクリックします。

詳細は、『Vivado Design Suite ユーザー ガイド: IP を使用した設計』(UG896) [参照 5] および『Vivado Design Suite ユーザー ガイド: 入門』(UG910) [参照 6] を参照してください。

注記: この章の図には Vivado 統合設計環境 (IDE) のスクリーンショットが使用されていますが、現在のバージョンとはレイアウトが異なる場合があります。

図 4-1 に、HDMI Transmitter Subsystem の [Customize IP] ダイアログ ボックスを示します。[Include HDCP 2.2 encryption] をオンにして、サブシステムに HDCP 2.2 IP のトランスミッターを含めます。

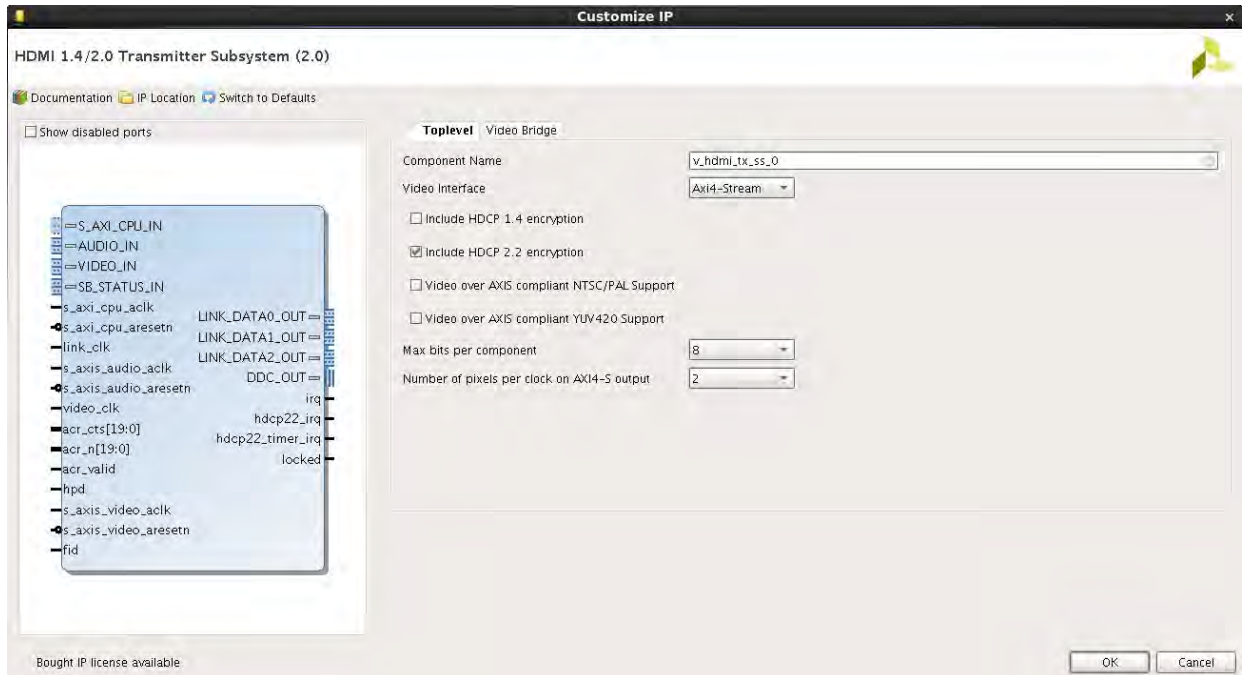


図 4-1: HDMI Transmitter Subsystem の HDCP 2.2 に関するカスタマイズ項目

図 4-2 に、HDMI Receiver Subsystem の [Customize IP] ダイアログ ボックスを示します。[Include HDCP 2.2 decryption] をオンにして、サブシステムに HDCP 2.2 IP のレシーバーを含めます。

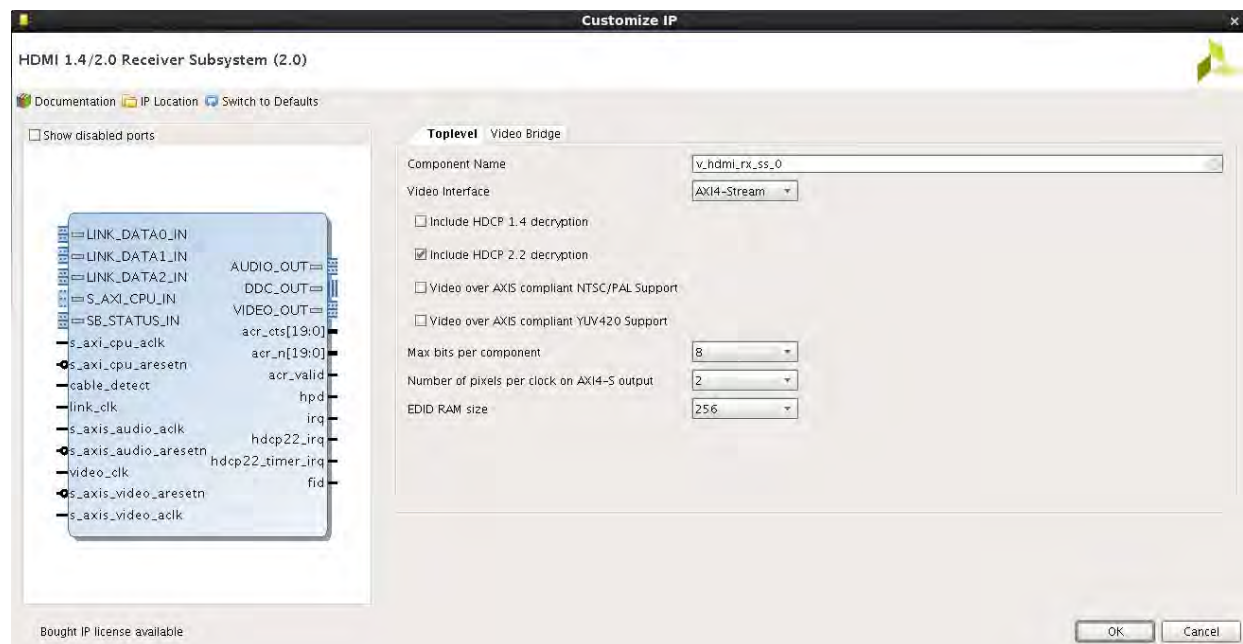


図 4-2: HDMI Receiver Subsystem の HDCP 2.2 に関するカスタマイズ項目

IP への制約

ここでは、Vivado Design Suite で IP に制約を指定する方法について説明します。

必須の制約

このセクションは、この IP には適用されません。

デバイス、パッケージ、スピード グレードの選択

このセクションは、この IP には適用されません。

クロック周波数

このセクションは、この IP には適用されません。

クロック管理

このセクションは、この IP には適用されません。

クロック配置

このセクションは、この IP には適用されません。

バンク設定

このセクションは、この IP には適用されません。

トランシーバーの配置

このセクションは、この IP には適用されません。

I/O 規格と配置

このセクションは、この IP には適用されません。

シミュレーション

Vivado シミュレーション コンポーネントについて、またサポートされているサードパーティ ツールについては、『Vivado Design Suite ユーザー ガイド: ロジック シミュレーション』(UG900) [\[参照 7\]](#) を参照してください。



重要: 7 シリーズまたは Zynq[®]-7000 デバイスをターゲットにした IP では、UNIFAST ライブラリはサポートされません。ザイリンクスの IP は UNISIM ライブラリでのみテストと認定が行われています。

合成およびインプリメンテーション

合成およびインプリメンテーションの詳細は、『Vivado Design Suite ユーザー ガイド: IP を使用した設計』(UG896) [\[参照 5\]](#) を参照してください。

アプリケーション ソフトウェア 開発

HDCP 2.2 トランスミッターおよびレシーバーには次の階層のドライバーがあります。

- HDCP 2.2 トランスミッター ドライバー (xhdcp22_tx.h)
 - HDCP 2.2 Cipher ドライバー (xhdcp22_cipher.h)
 - HDCP 2.2 Random Number Generator ドライバー (xhdcp22_rng.h)
 - HDCP 2.2 共通ユーティリティ (xhdcp22_common.h)
 - AXI Timer ドライバー (xtmrctr.h)
- HDCP 2.2 レシーバー ドライバー (hdcp22_rx.h)
 - HDCP 2.2 Cipher ドライバー (xhdcp22_cipher.h)
 - HDCP 2.2 Montgomery Modular Multiplier ドライバー (xhdcp22_mmult.h)
 - HDCP 2.2 Random Number Generator ドライバー (xhdcp22_rng.h)
 - HDCP 2.2 共通ユーティリティ (xhdcp22_common.h)
 - AXI Timer ドライバー (xtmrctr.h)

HDCP 2.2 ドライバーは、認証、ペアリング、局所性チェック、セッション キー交換、リンク インテグリティ チェック、トポロジ情報伝搬、およびストリーム管理情報伝搬を含む認証プロトコル ステート マシンを実装します。認証プロトコルに必要な暗号機能のほとんどはこれらのドライバーが実行しますが、RSA 復号化のみは性能要件を満たすために Montgomery Modular Multiplier によってハードウェアにオフロードされます。

デバイス ドライバー

以降のサブセクションでは、HDCP 2.2 トランスミッターおよびレシーバーのドライバー アーキテクチャ、およびドライバーの初期化と実行に使用する主な API について説明します。HDMI で HDCP 2.2 を使用する場合、HDMI ドライバーが HDCP 2.2 ドライバーのラッパーとしてユーザー アプリケーションに高次の抽象レイヤーを提供するため、ユーザー アプリケーションから HDCP 2.2 ドライバーを直接使用することはありません。したがって、このセクションの説明は参考にとどめてください。ただし、ドライバーの概念理解やデバッグにはこの情報が役立ちます。高次の HDCP ラッパー API の詳細は、『HDMI 1.4/2.0 Transmitter Subsystem v2.0 製品ガイド』(PG235) [参照 1] および『HDMI 1.4/2.0 Receiver Subsystem v2.0 製品ガイド』(PG236) [参照 2] を参照してください。

HDCP 2.2 認証プロトコルを処理するためのステート マシンは、これらドライバーによって実装されます。統合とデバッグを容易にするため、各ステート名は可能な限り仕様で定義された名前に揃えています。このステート マシンはオペレーティング システムに依存しないシンプルなスケジューラを使用します。メッセージとイベントを確実に処理するため、クライアント アプリケーション ソフトウェアは定期的にノンブロッキング ポーリング関数を呼び出します。HDCP 2.2 トランスミッターと HDCP 2.2 レシーバーはそれぞれマスターとスレーブの関係にあります。認証はトランスミッター側から開始し、レシーバーは要求があるまで待機します。

HDCP 2.2 認証のオクテット文字列を格納するバイト アレイは、ドライバー全体でビッグ エンディアンのバイト オーダーで定義されています。

HDCP 2.2 トランスミッター ドライバーの概要

HDCP 2.2 トランスミッターは、認証プロトコルにおいてマスターとして動作します。HDCP_HPD イベント発生後、トランスミッターは HDCP 2.2 対応レシーバーとの間でいつでも認証を開始できます。認証に成功した後、Cipher の暗号化を有効にするかどうかは HDCP 2.2 ドライバーではなくユーザー アプリケーション ソフトウェアで決定します。認証を再開すると、レシーバーとの同期のために Cipher カウンターがリセットされます。

HDCP 2.2 仕様では、レシーバーからのほとんどの応答メッセージに対してトランスミッターがタイムアウト チェックを実行することが要求されています。HDCP 2.2 トランスミッター ドライバーはハードウェア タイマーを使用してこのタイムアウト チェックを実行します。このタイマーは、メッセージ トランザクションの開始時に HDCP 2.2 トランスミッター ドライバーによって初期化されます。タイマーがタイムアウトすると割り込みイベントが生成され、割り込みサービスルーチンで処理が実行され、タイムアウトの発生を HDCP 2.2 トランスミッター ドライバーに通知します。

HDCP 2.2 トランスミッター ドライバーには、DDC メッセージの書き込みおよび読み出し トランザクション用に HDMI トランスミッター ドライバーに対してコールバックを設定する機能があります。HDMI トランスミッターは HDCP 2.2 トランスミッター ドライバーから物理 DDC インターフェイス経由で読み出し/書き込みコマンドを受信し、I2C インターフェイス経由で送信します。

HDCP 2.2 トランスミッター ドライバーの統合

このセクションでは、HDCP 2.2 トランスミッター ドライバーを初期化して実行するのに必要な手順について説明します。ドライバー インスタンスを確実に初期化するには、HDMI トランスミッター サブシステム ドライバーなどの高次ソフトウェアはここに示したのとほぼ同じ順番で関数を呼び出す必要があります。

1. HDCP 2.2 トランスミッター ドライバー インスタンスを初期化します。
 - XHdcp22Tx_LookupConfig
 - XHdcp22Tx_CfgInitialize
2. システム割り込みコントローラーに HDCP 2.2 トランスミッターのタイマー割り込みを接続します。タイマーは、接続した HDCP 2.2 レシーバーの応答のタイミング違反を防ぐため、およびログにタイムスタンプを記録するために使用します。次の関数からは、HDCP 2.2 トランスミッター サブシステム内にあるタイマー インスタンスが返されます。
 - XHdcp22Tx_GetTimer
3. DDC 読み出しおよび書き込みハンドラーに対するコールバックを設定します。
 - XHdcp22Tx_SetCallbacks
4. DCP グローバル定数を Cipher にロードします。
 - XHdcp22Tx_LoadLc128
5. SRM テーブルをドライバーにロードします。
 - XHdcp22Tx_LoadRevocationTable
6. ステート マシンがレシーバーからのメッセージをいつポーリングするか(すなわち期待されるメッセージの到着をチェック開始するまでのタイムアウト時間)を決定するポーリング値を設定します。ポーリングが必要ない場合は、値 0 を使用します。それ以外の整数値は、いつポーリングを開始するかを指定します。これは、 $100 - (100/\text{value})\%$ として計算します。値が 1 の場合、ポーリングがただちに開始します。値が 3 の場合、タイムアウトまでの時間の 66% が経過するとポーリングが開始します。デフォルトの値は 4 (75%) に設定されます。
 - XHdcp22Tx_SetMessagePollingValue
7. リピーター モードを設定します。
 - XHdcp22Tx_SetRepeater
8. 内部ステート マシンをリセットして有効にします。
 - XHdcp22Tx_Reset
 - XHdcp22Tx_Enable

9. 次の関数を呼び出してレシーバーとの間で認証を開始します。認証は、レシーバーからホット プラグ検出 (HPD) イベントを受信した後にユーザーが開始する必要があります。
 - XHdcp22Tx_Authenticate
10. HDCP 2.2 トランスミッターの内部ステート マシンを正しく動作させるには、ポーリング関数を定期的呼び出す必要があります。この関数呼び出しはユーザー アプリケーションのメイン ループに挿入し、HDCP 2.2 が有効な間は常時実行するようにします。このポーリング関数の戻り値を使用して、現在の認証ステートを検出できます。認証に成功した後も、接続した HDCP 2.2 レシーバーからの再認証要求を検出するためにポーリングを継続する必要があります。
 - XHdcp22Tx_Poll
11. 認証ステータスは、次の関数でチェックできます。
 - XHdcp22Tx_IsInProgress
 - XHdcp22Tx_IsAuthenticated
12. 暗号化は、認証に成功した後いつでも有効または無効にできます。HDCP 2.2 トランスミッターはセッション キーを使用して Cipher を初期化しますが、暗号化を有効/無効にはしません。暗号化はユーザー アプリケーションで有効にします。なお、暗号化を有効にできるのは接続した HDCP 2.2 レシーバーの認証に成功した場合のみです。暗号化を有効または無効にするには次の関数を使用します。
 - XHdcp22Tx_EnableEncryption
 - XHdcp22Tx_DisableEncryption
13. リピーターのトポロジ情報を取得します。
 - XHdcp22Tx_GetTopologyField
 - XHdcp22Tx_GetTopologyReceiverIdList

HDCP 2.2 トランスミッターのコールバック イベント

HDCP 2.2 トランスミッター ドライバーにはコールバック関数があり、ユーザーはこれらを登録してステート マシンの重要な遷移やイベント発生時にタスクを実行できます。これらのコールバックは、リピーター トポロジおよびストリーム管理の情報を伝搬するのに不可欠な役割を果たします。次に、コールバックの一覧を示します。

- XHDCP22_TX_HANDLER_AUTHENTICATED
ステート マシンが Authenticated ステートに遷移すると実行されます。
- XHDCP22_TX_HANDLER_UNAUTHENTICATED
ステート マシンがいずれかのステートから Unauthenticated ステートに遷移すると実行されます。
- XHDCP22_TX_HANDLER_DOWNSTREAM_TOPOLOGY_AVAILABLE
ダウンストリーム デバイスからのトポロジ情報が存在することをステート マシンが検出すると実行されます。ダウンストリーム デバイスがエンドポイント レシーバーの場合、トポロジ情報には 1 つのデバイスしか含まれません。ダウンストリーム デバイスがリピーターの場合、トポロジ情報には最大 31 のデバイスが含まれます。トポロジ情報は、関数 XHdcp22Tx_GetTopologyField と XHdcp22Tx_GetTopologyReceiverIdList を使用して抽出します。

HDCP 2.2 トランスミッターの割り込みと性能

サブシステムによって生成された割り込みは、HDCP 2.2 トランスミッター ドライバーが処理します。次に、割り込みソースおよび関連するイベントの一覧を示します。

1. HDCP 2.2 Timer 割り込み
 - Timer0 イベント: 未使用。将来使用するために予約。
 - Timer1 イベント: カウントダウン タイマーがタイムアップしたことを示す。このイベントを使用してプロトコルのタイムアウトを判定する。
2. HDCP 2.2 Cipher 割り込み
 - 未使用。将来使用するために予約。

メッセージのタイムアウト間隔は HDCP 2.2 トランスミッターが計測して応答します。HDCP 2.2 プロトコルの各種ステートに対するタイムアウト要件の詳細は、『High-bandwidth Digital Content Protection System—Mapping HDCP to HDMI Revision 2.2』[参照 12] を参照してください。HDCP 2.2 トランスミッターのステート マシンは XHdcp22Tx_Poll 関数を使用して動作するため、特に認証の実行中などではこの関数に十分な CPU 実行時間が割り当てられるように注意する必要があります。認証中にこのポーリング関数の実行が遅れた場合、トランスミッターによってプロトコル タイミング違反が検出され、認証に失敗することがあります。プロトコルのタイミング性能はログ バッファに挿入されたタイムスタンプに基づいてチェックできます。詳細は「HDCP 2.2 ドライバーのログ機能」を参照してください。HDCP 2.2 プロトコルで最もタイミング要件が厳しいのは局所性チェックで、レシーバーは 20ms 以内にメッセージを受信する必要があります。

HDCP 2.2 レシーバー ドライバーの概要

HDCP 2.2 レシーバーは認証プロトコルでスレーブとして動作し、トランスミッターが認証を開始するまで待機します。レシーバー ドライバーは、ドライバーが初期化されると HDCP 2.2 機能ビットをセットし、トランスミッターに HDCP 2.2 トランザクションを開始してよいことを通知します。

認証プロトコルの各フェーズ間のタイミング遅延は HDCP 2.2 レシーバーではなくトランスミッターが追跡します。このため、HDCP 2.2 レシーバーは認証プロトコル中にプリエンティブにタイムアウトせず、トランスミッター側で違反を検出して認証を再開します。ただし例外として、レシーバー ドライバーは RxStatus レジスタの READY ビットがアサートされてから 2 秒以内に RepeaterAuth_Send_Ack メッセージを受信したかどうかをチェックします。このメッセージを受信できていない場合のみ、ステート マシンはプリエンティブに認証を中止します。認証中に予期しないメッセージを受信またはエラー条件を検出した場合、HDCP 2.2 レシーバーはメッセージ バッファをクリアし、Unauthenticated ステートに遷移 (リセット) します。認証初期化メッセージを受信すると、レシーバーはそれまでの認証プロセスのステート情報をすべてクリアします。

認証を実行するには、マスター キーの復号化に使用する RSA 秘密キーが必要です。HDCP 2.2 レシーバーには、認証およびキー交換中に Cipher にプログラムされる秘密キー ポインターと DCP グローバル定数をドライバーにロードする機能があります。HDCP 2.2 レシーバー ドライバーにはキー管理 (キーの格納と取り出し) の機能はないため、ユーザー アプリケーションで実行する必要があります。HDCP 2.2 仕様の要求に従って、秘密キーの完全性と機密性をユーザー アプリケーションで維持する必要があります。

秘密キーをロードする以外に、HDCP 2.2 ドライバーには証明書送信 (AKE_Send_Cert) メッセージの一部として送信される公開証明書をロードする API もあります。公開キー証明書はレシーバー デバイスごとに異なり、DCP によって生成される署名を含みます。公開証明書は機密性が不要とされず、認証要求ごとにトランスミッターに提供されます。

レシーバー認証プロトコルで必要な暗号機能は、マスター キーの復号化に使用する RSA 復号化以外はソフトウェアで実装されます。RSA 復号化ではべき剰余演算を実行しますが、復号化指数のサイズが大きいため大量の演算が必要となります。したがって、タイミング要件を満たすためにモンゴメリ剰余乗算アルゴリズムはハードウェアで実行します。

HDCP 2.2 レシーバー ドライバーには、DDC メッセージの書き込みおよび読み出し トランザクション用に HDMI レシーバー ドライバーに対してコールバックを設定する機能があります。HDMI レシーバーは HDCP 2.2 レシーバー ドライバーから物理 DDC インターフェイス経由で読み出し/書き込みコマンドを受信し、I2C インターフェイス経由で送信します。

HDCP 2.2 レシーバードライバの統合

このセクションでは、HDCP 2.2 レシーバードライバを初期化して実行するのに必要な手順について説明します。ドライバ インスタンスを確実に初期化するには、HDMI レシーバ サブシステム ドライバなどの高次ソフトウェアはここに示したのとほぼ同じ順番で関数を呼び出す必要があります。

1. HDCP 2.2 トランスミッタードライバ インスタンスを初期化します。
 - XHdcp22Rx_LookupConfig
 - XHdcp22Rx_CfgInitialize
2. DDC 読み出しおよび書き込みハンドラーに対するコールバックを設定します。
 - XHdcp22Rx_SetCallbacks
3. DCP グローバル定数を Cipher にロードします。
 - XHdcp22Rx_LoadLc128
4. レシーバの公開証明書へのポインタをドライバ インスタンスにロードします。
 - XHdcp22Rx_LoadPublicCert
5. レシーバの秘密キーへのポインタをドライバ インスタンスにロードします。この関数は、秘密キーのパラメータを使用してモンゴメリ乗算の定数を計算します。
 - XHdcp22Rx_LoadPrivateKey
6. メッセージ処理のために DDC コールバック関数を HDMI レシーバに接続します。これらの関数は、HDMI レシーバの書き込みバッファに完全なメッセージが格納された場合、または HDMI レシーバの読み出しバッファから完全なメッセージが読み出された場合に呼び出されます。
 - XHdcp22Rx_SetWriteMessageAvailable
 - XHdcp22Rx_SetReadMessageComplete
7. エラー処理のために DDC コールバック関数を HDMI Receiver に接続します。DDC メッセージトランザクションに失敗すると DDC エラーがセットされます。HDMI レシーバが ECC パリティ エラーを 50 個連続して検出すると、リンク エラーがセットされます。
 - XHdcp22Rx_SetDdcError
 - XHdcp22Rx_SetLinkError
8. リピーター モードを設定します。
 - XHdcp22Rx_SetRepeater
9. 内部ステート マシンをリセットして有効にします。
 - XHdcp22Rx_Reset
 - XHdcp22Rx_Enable
10. HDCP 2.2 レシーバの内部ステート マシンを正しく動作させるには、ポーリング関数を定期的呼び出す必要があります。この関数呼び出しはユーザー アプリケーションのメイン ループに挿入し、HDCP 2.2 が有効な間は常時実行するようにします。認証に成功した後も、リンク インテグリティチェックを実行するためにポーリングを継続する必要があります。
 - XHdcp22Rx_Poll
11. 認証ステータスは、次の関数でチェックできます。
 - XHdcp22Rx_IsInProgress
 - XHdcp22Rx_IsAuthenticated
12. リピータのトポロジ情報を設定します。
 - XHdcp22Rx_SetTopologyField
 - XHdcp22Rx_SetTopoogyReceiverIdList
 - XHdcp22Rx_SetTopologyUpdate

HDCP 2.2 レシーバーのコールバック イベント

HDCP 2.2 レシーバー ドライバーにはコールバック関数があり、ユーザーはこれらを登録してステート マシンの重要な遷移やイベント発生時にタスクを実行できます。これらのコールバックは、リピーター トポロジおよびストリーム管理の情報を伝搬するのに不可欠な役割を果たします。次に、コールバックの一覧を示します。

- **XHDCP22_RX_HANDLER_AUTHENTICATED**
ステート マシンが **Authenticated** ステートに遷移すると実行されます。
- **XHDCP22_RX_HANDLER_UNAUTHENTICATED**
ステート マシンが **Unauthenticated** ステートに遷移すると実行されます。
- **XHDCP22_RX_HANDLER_AUTHENTICATION_REQUEST**
認証要求 (**AKE_Init**) を受信すると実行されます。
- **XHDCP22_RX_HANDLER_TOPOLOGY_UPDATE**
レシーバーがセッション キーを交換し、アップストリームに伝搬するレシーバー ID リストの受信および収集を待機しているときに実行されます。
- **XHDCP22_RX_HANDLER_STREAM_MANAGE_REQUEST**
ストリーム管理要求を受信すると実行されます。ストリーム タイプを抽出するには、**XHdcp22Rx_GetContentStreamType** 関数を使用します。
- **XHDCP22_RX_HANDLER_ENCRYPTION_UPDATE**
Cipher が受信するコンテンツの暗号化ステータスが **Unencrypted** から **Encrypted** (または **Encrypted** から **Unencrypted**) へ変化すると実行されます。Cipher の暗号化が有効かどうかを調べるには、**XHdcp22Rx_IsEncryptionEnabled** 関数を使用します。このコールバックは、ステート マシンが **Authenticated** ステートに遷移した後、1 秒ごとに実行されます。リピーター モードが有効な場合は、トポロジ伝搬によって最大 3 秒の間隔が存在し、レシーバーに通知されないまま暗号化されたコンテンツが送信される可能性があります。

HDCP 2.2 レシーバーの割り込みと性能

サブシステムによって生成された割り込みは、HDCP 2.2 レシーバー ドライバーが処理します。次に、割り込みソースおよび関連するイベントの一覧を示します。

1. HDCP 2.2 Timer 割り込み
 - **Timer0** イベント: 未使用。
 - **Timer1** イベント: カウントダウン タイマーがタイムアップしたことを示す。このイベントを使用してプロトコルのタイムアウトを判定する。
2. HDCP 2.2 Cipher 割り込み
 - 未使用。将来使用するために予約。

HDCP 2.2 プロトコルの各種ステートに対するタイムアウト要件の詳細は、『High-bandwidth Digital Content Protection System—Mapping HDCP to HDMI Revision 2.2』[参照 12] を参照してください。HDCP 2.2 レシーバーのステート マシンは **XHdcp22Rx_Poll** 関数を使用して動作するため、特に認証の実行中などではこの関数に十分な CPU 実行時間が割り当てられるように注意する必要があります。認証中にこのポーリング関数の実行が遅れた場合、トランスミッターによってプロトコル タイミング違反が検出され、認証に失敗することがあります。プロトコルのタイミング性能はログ バッファに挿入されたタイムスタンプに基づいてチェックできます。詳細は「[HDCP 2.2 ドライバーのログ機能](#)」を参照してください。HDCP 2.2 プロトコルで最もタイミング要件が厳しいのは局所性チェックで、レシーバーは 20ms 以内にメッセージを受信する必要があります。

HDCP 2.2 ドライバーのログ機能

HDCP 2.2 ドライバーにはパフォーマンス特性評価およびデバッグ用のログ バッファがあります。ログ イベントは循環形式のログ バッファに記録され、ユーザー コマンドによって出力されます。すべてのログ イベントには、ハードウェア タイマーによってタイムスタンプが挿入されます。このタイマーは 32 ビットのため、100MHz クロックを使用した場合 42.9 秒後にラップアラウンドします。ユーザー アプリケーションからログ バッファへのインターフェイスとして、次の関数を使用できます。

1. 内部ログ バッファをクリアしてタイマーを再スタートするには次の関数を使用します。オプションで `verbose` パラメータを使用すると、プロトコルの詳細情報もログに記録されます。

- `XHdcp22Tx_LogReset`
- `XHdcp22Rx_LogReset`

2. ログ バッファの内容を表示してクリアするには次の関数を使用します。ログは次のフォーマットで表示されます。

`[<relative_time_in_us> : <delta_time_in_us>] <log_message>`

- `XHdcp22Tx_LogDisplay`
- `XHdcp22Rx_LogDisplay`

3. 事前定義済みのログ イベント以外にオプションでユーザー レベルのログ イベントがあり、次の関数を使用してカスタム ログ メッセージをログ バッファに挿入できます。

- `XHdcp22Tx_LogWr`
- `XHdcp22Rx_LogWr`

アップグレード

この付録は、このコアのリリースには適用されません。

デバッグ

この付録では、ザイリンクス サポート ウェブサイトより入手可能なリソースおよびデバッグ ツールについて説明します。



ヒント: IP の生成にエラーが発生し停止した場合、ライセンスに問題がある可能性があります。詳細は、[第 1 章の「ライセンスチェッカー」](#)を参照してください。

ザイリンクス ウェブサイト

HDCP 2.2 IP サブシステムを使用した設計およびデバッグでヘルプが必要な場合は、[ザイリンクス サポート ウェブページ](#)から製品の資料、リリース ノート、アンサーなどを参照するか、テクニカル サポートでサービス リクエストを作成してください。

資料

この製品ガイドは HDCP 2.2 IP サブシステムに関する主要資料です。このガイド、並びに設計プロセスで使用する各製品の関連資料はすべて、ザイリンクス サポート ウェブ ページ (<https://japan.xilinx.com/support>) または Xilinx Documentation Navigator から入手できます。

Xilinx Documentation Navigator は、[ダウンロード ページ](#)からダウンロードできます。このツールの詳細および機能は、インストール後にオンライン ヘルプを参照してください。

アンサー

アンサーには、よく発生する問題についてその解決方法、およびザイリンクス製品に関する既知の問題などの情報が記載されています。アンサーは、ユーザーが該当製品の最新情報にアクセスできるよう作成および管理されています。

このコアに関するアンサーの検索には、[ザイリンクス サポート ウェブ ページ](#)にある検索ボックスを使用します。よりの確な検索結果を得るには、次のようなキーワードを使用してください。

- 製品名
- ツールで表示されるメッセージ
- 問題の概要

検索結果は、フィルター機能を使用してさらに絞り込むことができます。

HDCP 2.2 IP サブシステムに関するマスター アンサー

AR: [66762](#)

テクニカル サポート

ザイリンクスは、製品資料の説明に従って使用されているこの IP に対するテクニカル サポートを[ザイリンクス サポート ウェブ ページ](#)で提供しています。ただし、次に該当する場合、タイミング、機能、サポートは保証されません。

- 資料で定義されていないデバイスにソリューションをインプリメントした場合。
- 資料で定義されている許容範囲を超えてカスタマイズした場合。
- 「DO NOT MODIFY」とされているデザイン セクションに変更を加えた場合。

ザイリンクス テクニカル サポート へのお問い合わせは、[ザイリンクス サポート ウェブ ページ](#)を参照してください。

デバッグ ツール

HDCP 2.2 IP サブシステム デザインの問題を解決するには、数多くのツールを利用できます。さまざまな状況をデバッグするのに有益なツールを理解しておくことが重要です。

Vivado Design Suite のデバッグ機能

Vivado® Design Suite のデバッグ機能は、Logic Analyzer および Virtual I/O コアをユーザー デザインに直接挿入します。デバッグ機能を使用すると、トリガー条件を設定して、アプリケーションおよび統合ブロックのポート信号をハードウェアに取り込むことができます。取り込まれた信号は、その後解析できます。Vivado IDE のこの機能は、ザイリンクス デバイスで実行されるデザインの論理デバッグおよび検証に使用されます。

Vivado ロジック解析は次の IP ロジック デバッグ コアと共に使用されます。

- ILA 2.0 (およびそれ以降のバージョン)
- VIO 2.0 (およびそれ以降のバージョン)

詳細は、『Vivado Design Suite ユーザー ガイド: プログラムおよびデバッグ』(UG908) [参照 9] を参照してください。

ハードウェア デバッグ

ハードウェアの問題は、リンク立ち上げ時の問題から、テスト後に生じる問題までさまざまです。ここでは、一般的な問題のデバッグ手順を説明します。Vivado のデバッグ機能は、ハードウェア デバッグに有益なリソースです。次の各セクションに示す信号を Vivado のデバッグ機能でプローブすることで、個々の問題をデバッグできます。

一般的なチェック

IP に対するタイミング制約がサンプル デザインからすべて適切に取り込まれていること、さらにインプリメンテーション時にこれらの制約がすべて満たされていることを確認します。

- 配置配線後のタイミング シミュレーションで正しく動作しているかを確認します。タイミング シミュレーションでは発生しない問題がハードウェアで発生する場合、PCB の問題である可能性があります。すべてのクロック ソースがアクティブでクリーンであることを確認してください。
- デザインで MMCM を使用している場合、locked ポートをモニターして、すべての MMCM がロックしていることを確認します。
- 出力が 0 になった場合は、ライセンスを確認してください。

HDCP 2.2 に関するチェック

このセクションでは、HDCP 2.2 IP のトランスミッターおよびレシーバーをユーザー アプリケーションに統合する際に発生する一般的な問題について説明します。デバッグの前に、付録 A 「アプリケーション ソフトウェア開発」を参照してドライバー初期化シーケンスを理解しておく必要があります。詳細はください。

HDCP 2.2 認証に失敗する場合、いくつかの理由が考えられます。次に、一般的なチェック項目を示します。

- ケーブルは正しく接続されていますか。ケーブルが接続されていない場合やケーブル不良の場合、認証に失敗します。
- ステート マシンが有効にされ、動作していますか。ドライバーを初期化した後も、ユーザーが明示的にステート マシンを有効にして実行する必要があります。
- ソフトウェアのスタック オーバーフローが発生していませんか。スタックがオーバーランしない十分なサイズを確保してください。スタック オーバーフローが発生すると、予期しない動作となることがあります。
- 認証エラーはどこで発生していますか。ログ バッファを調べてエラーの発生箇所を特定してください。ログ バッファにはタイムスタンプが挿入されているため、メッセージトランザクションの間隔を調べることができます。ログ バッファにはエラー イベントも記録されるため、デバッグに役立ちます。タイムアウトによるエラーが発生している場合、プロセッサ負荷が大きいことが原因として考えられます。
- キーは正しくロードされていますか。タイムアウトによって認証に失敗する場合、公開証明書と秘密キーがレシーバーのドライバーに正しくロードされているかどうか確認してください。これらのキーが正しくロードされていないと認証に失敗します。

ターゲットとなるビデオ プロトコル サブシステムに応じたキーの正しいロード方法の詳細は、『HDMI 1.4/2.0 Receiver Subsystem v2.0 製品ガイド』(PG236) [参照 2] を参照してください。

HDCP 2.2 認証に成功してもスノー ノイズしか表示されない場合は、次の項目を確認してください。

- DCP グローバル定数は正しくロードされていますか。グローバル定数を正しくロードしていないとノイズが発生します。
- システムをハード リセットして問題が解決するかどうか確認してください。
- それでもノイズが解消されない場合、トランスミッターまたはレシーバーでの同期データの暗号化に問題があることが考えられます。問題を切り分けて、判明した内容を報告してください。

その他のリソースおよび法的通知

ザイリンクス リソース

アンサー、資料、ダウンロード、フォーラムなどのサポート リソースは、[ザイリンクス サポート サイト](#)を参照してください。

参考資料

次の資料は、この製品ガイドの補足資料として役立ちます。

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. 『HDMI 1.4/2.0 Transmitter Subsystem v2.0 製品ガイド』(PG235: [英語版](#)、[日本語版](#))
2. 『HDMI 1.4/2.0 Receiver Subsystem v2.0 製品ガイド』(PG236: [英語版](#)、[日本語版](#))
3. 『Kintex UltraScale FPGA GTH トランシーバーを使用した HDMI 2.0 の実装』(XAPP1275: [英語版](#)、[日本語版](#))
4. 『Vivado Design Suite ユーザー ガイド: IP インテグレーターを使用した IP サブシステムの設計』(UG994: [英語版](#)、[日本語版](#))
5. 『Vivado Design Suite ユーザー ガイド: IP を使用した設計』(UG896: [英語版](#)、[日本語版](#))
6. 『Vivado Design Suite ユーザー ガイド: 入門』(UG910: [英語版](#)、[日本語版](#))
7. 『Vivado Design Suite ユーザー ガイド: ロジック シミュレーション』(UG900: [英語版](#)、[日本語版](#))
8. 『ISE から Vivado Design Suite への移行ガイド』(UG911: [英語版](#)、[日本語版](#))
9. 『Vivado Design Suite ユーザー ガイド: プログラムおよびデバッグ』(UG908: [英語版](#)、[日本語版](#))
10. 『Vivado Design Suite ユーザー ガイド: インプリメンテーション』(UG904: [英語版](#)、[日本語版](#))
11. 『LogiCORE IP AXI Interconnect 製品ガイド』([PG059](#))
12. 『HDCP 2.2 on HDMI Specification』(『High-bandwidth Digital Content Protection System—Mapping HDCP to HDMI Revision 2.2』) ([Digital Content Protection のウェブサイト](#))
13. 『Errata to HDCP 2.2 on HDMI Specification』([Digital Content Protection のウェブサイト](#))

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10月5日	1.0	全般: <ul style="list-style-type: none"> リピーターおよびコンバーターの動作に関する情報を追加。 「IP を使用するデザイン」 ユーザー アプリケーションに必要なキーのリストを更新。 「一般的なデザイン ガイドライン」のセクションを追加。 付録「アプリケーション ソフトウェア開発」 手順を全面的に更新。 「HDCP 2.2 トランスミッターのコールバック イベント」および「HDCP 2.2 レシーバーのコールバック イベント」のセクションを追加。
2016年4月6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、<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になります。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 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。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。