



ハイブリッド・メモリー・キューブ・コントローラー IP コア・ユーザーガイド

インテル® Quartus® Prime 開発デザインスイートの更新情報: **Quartus Prime Pro – Stratix 10 Edition Beta**



UG-01152 | 2016.08.08

最新版をウェブからダウンロード: [PDF](#) | [HTML](#)

目次

1 ハイブリッド・メモリー・キューブ・コントローラー IP コアについて	4
1.1 HMC コントローラー IP コアでサポートされる機能.....	5
1.2 HMC コントローラー IP コアがサポートする HMC トランザクションのタイプ.....	6
1.3 デバイスファミリー・サポート.....	7
1.4 IP コアの検証.....	8
1.4.1 シミュレーション.....	8
1.5 性能およびリソース使用率.....	9
1.6 デバイスのスピードグレードのサポート.....	9
1.7 リリース情報.....	10
2 HMC コントローラー IP コアのスタート	11
2.1 IP コアのライセンス取得.....	11
2.1.1 OpenCore Plus IP 評価機能.....	12
2.2 IP コアのパラメーターとオプションの指定.....	12
2.3 HMC コントローラー IP コアのパラメーター.....	13
2.3.1 RX Mapping パラメーターおよび TX Mapping パラメーター.....	15
2.4 インテル FPGA IP コア向けに生成されるファイル.....	18
2.5 デザインへの IP コアの統合.....	19
2.5.1 ピンの制約.....	19
2.5.2 必要な外部インターフェイス.....	20
2.6 インテル FPGA IP コアのシミュレーション.....	25
3 機能概要	27
3.1 上位レベルのブロック図.....	27
3.2 インターフェイスの概要.....	28
3.2.1 アプリケーション・インターフェイス.....	28
3.2.2 HMC インターフェイス.....	28
3.2.3 外部 I ² C マスターへのインターフェイス.....	28
3.2.4 コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイス.....	29
3.2.5 ステータス・インターフェイスおよびデバッグ・インターフェイス.....	29
3.2.6 トランシーバー・コントロール・インターフェイス.....	29
3.3 クロッキング構造.....	30
3.4 初期化とリセット.....	31
3.5 M20K ECC サポート.....	34
3.6 フロー制御.....	34
3.7 エラー検出および管理.....	35
3.8 テスト機能.....	36
4 HMC コントローラー IP コア信号	37
4.1 アプリケーション・インターフェイス信号.....	37
4.1.1 アプリケーション要求インターフェイス.....	37
4.1.2 アプリケーション応答インターフェイス.....	41
4.1.3 HMC コントローラー IP コアのデータパス例.....	44
4.2 HMC インターフェイス信号.....	44
4.3 I ² C マスターへのインターフェイス上の信号.....	46



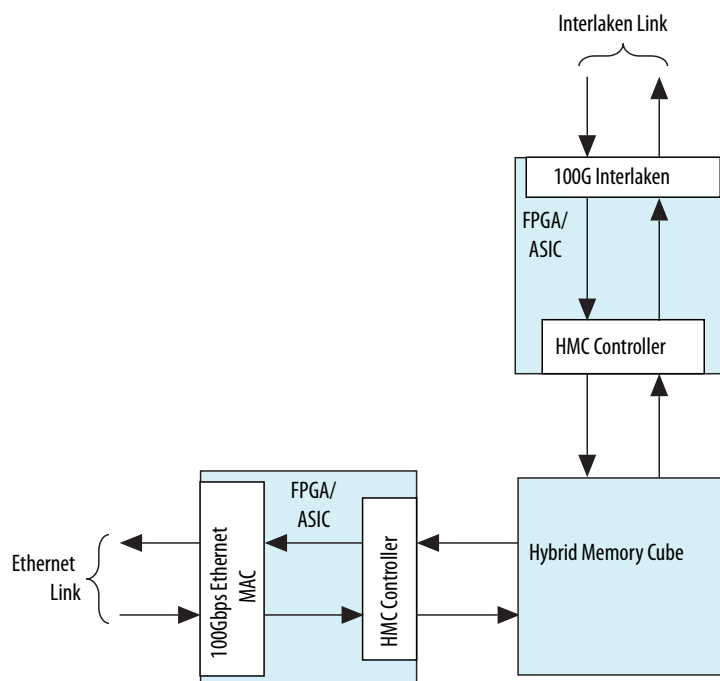
4.4	コントロール・レジスター・インターフェイス信号およびステータス・レジスター・インターフェイス信号.....	47
4.5	ステータス信号およびデバッグ信号.....	48
4.6	クロック信号およびリセット信号.....	48
4.7	トランシーバー・リコンフィグレーション信号.....	49
4.8	外部 PLL へのインターフェイス上の信号.....	50
5	HMC コントローラー IP コアのレジスターマップ.....	51
5.1	CONTROL レジスター.....	52
5.2	XCVR_STATUS レジスター.....	53
5.3	LANE_STATUS レジスター.....	53
5.4	LINK_STATUS レジスター.....	53
5.5	ERROR_RESPONSE レジスター.....	54
5.6	LIMIT_OUTSTANDING_PACKETS レジスター.....	54
5.7	割り込み関連のレジスター.....	55
5.8	エラーおよびリトライ・スタティック・レジスター.....	58
6	HMC コントローラー IP コア Stratix 10 のデザイン例.....	60
A	HMC コントローラー IP コア・ユーザーガイド・アーカイブ.....	61
B	HMC コントローラー IP ユーザーガイドの資料改訂履歴.....	62
B.1	HMC コントローラー IP ユーザーガイド改訂履歴.....	62

1 ハイブリッド・メモリー・キューブ・コントローラー IP コアについて

ハイブリッド・メモリー・キューブ (HMC) 仕様は、既存のメモリー・アーキテクチャーよりも帯域幅および電力効率を大幅に向上させる新しいタイプのメモリーデバイスを定義しています。HMC 仕様は、高性能コンピューターと次世代ネットワーク機器を対象としており、幅広いアプリケーションにスケーラビリティを提供します。

インテル® HMC コントローラー IP コアは外部 HMC デバイスに容易にアクセスすることができます。HMC デバイスは、小型フォームファクターで大量のメモリーに高帯域幅で信頼性の高いアクセスを提供し、高性能のメモリー集約型アプリケーションで大幅なシステムコスト削減を実現します。HMC コントローラー IP コアは、外部 HMC デバイスと通信して設計に帯域幅とパフォーマンスを組み込むことができる、シンプルなユーザー・インターフェイスを提供します。

図 -1: 一般的な HMC コントローラー・アプリケーション



関連情報

- 61 ページの [HMC コントローラー IP コア・ユーザーガイド・アーカイブ](#)
- [ハイブリッド・メモリー・キューブ・コントローラー・デザイン例のユーザーガイド](#)
- [インテル FPGA IP コアについて](#)
すべてのインテル FPGA IP コアについて、パラメーター化、生成、アップグレード、および IP コアのシミュレーションといった基本的な情報を提供します。



- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトおよび IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。
- [Hybrid Memory Cube Specification 1.1](#)
HMC 仕様は、ハイブリッド・メモリー・キューブ・コンソーシアム (HMCC) のウェブページからダウンロードできます。

1.1 HMC コントローラー IP コアでサポートされる機能

インテル HMC コントローラー IP コアは以下の機能を備えています。

- *Hybrid Memory Cube Specification 1.1* に準拠した外部 HMC デバイス搭載のインテル FPGA 高速トランシーバーを経由した通信機能
- 10 Gbps、12.5 Gbps、または 15 Gbps の各レーンレートでの HMC デバイスとの通信機能
- コントロール・レジスターおよびステータスレジスターにアクセスするための Avalon[®] メモリーマップド (Avalon-MM) インターフェイス機能
- 1 ~ 4 つの単一 512 ビットのクライアント・データ・インターフェイスを備える HMC デバイスの 16 レーンへの接続。複数のデータ・インターフェイスにより HMC リンクの使用率が向上する。
- すべての有効なペイロードサイズでメモリーの READ および WRITE トランザクションのサポート
- ATOMIC トランザクション、BIT WRITE トランザクション、および WRITE トランザクションのポストドおよびノンポストド・バージョンのサポート
- MODE READ および MODE WRITE トランザクションのサポート
- IP コアが要求を受け取った順に各アプリケーション応答インターフェイスで応答を送信するオプションの応答のリオーダーリングのサポート。このオプションを選択すると、IP コアがタグを管理し、タグはクライアント・インターフェイスでは表示されない。
- デバイスのリソース要件を低減するための受信 (RX) リソースフロー制御での Response Open Loop モードのサポート
- トークンベースの送信 (TX) フロー制御のサポート
- ポイズンドパケットのサポート
- 柔軟なボード設計のためのトランシーバー・レーンのリオーダーリングのサポート
- リンク・トレーニング・シーケンスのサポート、ならびに、ワード・アライメント、レーン・アライメントおよびトランシーバーのステータス情報をリアルタイムで表示
- 高速シミュレーション・サポートの提供
- リアルタイムのエラー統計の提供
- ハードウェアとソフトウェアのリセット制御の提供
- 消費電力制御の提供
- PHY 信号の整合性をデバッグまたは監視のための System Console を介するトランシーバー・レジスターへの ADME ダイレクトアクセスのオプション・サポート
- IP コアでコンフィグレーションされたすべての M20K メモリーブロックに ECC サポートを含めるためのオプション提供

設計で HMC デバイスへのマルチリンク接続をサポートするには、複数の HMC コントローラー IP コアが個別の HMC リンクを介して同じ HMC デバイスと通信するようにコンフィグレーションします。各 HMC コントローラー IP コアは、単一 HMC デバイスリンクに接続します。

詳細な HMC 仕様については、*Hybrid Memory Cube Specification 1.1* を参照してください。

関連情報

[Hybrid Memory Cube Specification 1.1](#)

HMC 仕様は、ハイブリッド・メモリー・キューブ・コンソーシアム (HMCC) のウェブページからダウンロードできます。

1.2 HMC コントローラー IP コアがサポートする HMC トランザクションのタイプ

インテル HMC コントローラー IP コアは、すべての HMC トランザクションをサポートしています。

HMC コントローラーから HMC デバイス・パケット・タイプへ

HMC インターフェイス上の転送は、128 ビット・フローユニット (FLIT) のシーケンスです。HMC コントローラー IP コアは、HMC デバイスへのリンク上で次のパケットタイプを生成します。

- NULL FLIT
- ポインターリターン (PRET) (シングル FLIT パケット)
- Init リトライ (IRTRY) (シングル FLIT パケット)
- READ 要求 (シングル FLIT パケット)
- 16 バイトの WRITE またはポストド WRITE 要求 (2-FLIT パケット)
- 32 バイトの WRITE またはポストド WRITE 要求 (3-FLIT パケット)
- 48 バイトの WRITE またはポストド WRITE 要求 (4-FLIT パケット)
- 64 バイトの WRITE またはポストド WRITE 要求 (5-FLIT パケット)
- 80 バイトの WRITE またはポストド WRITE 要求 (6-FLIT パケット)
- 96 バイトの WRITE またはポストド WRITE 要求 (7-FLIT パケット)
- 112 バイトの WRITE またはポストド WRITE 要求 (8-FLIT パケット)
- 128 バイトの WRITE またはポストド WRITE 要求 (9-FLIT パケット)
- BIT WRITE またはポストド BIT WRITE 要求 (2-FLIT パケット)
- MODE READ 要求 (シングル FLIT パケット)
- MODE WRITE 要求 (2-FLIT パケット)
- デュアル 8 バイト ADD IMMEDIATE またはポストドデュアル 8 バイト ADD IMMEDIATE 要求 (2-FLIT パケット)
- シングル 16 バイト ADD IMMEDIATE またはポストドシングル 16 バイト ADD IMMEDIATE 要求 (2-FLIT パケット)

HMC コントローラー IP コアは Response Open Loop モードで動作するため、Token Return (TRET) パケットを生成しません。



HMC デバイスから HMC コントローラー・パケット・タイプへ

HMC コントローラー IP コアは、HMC デバイスにより生成された以下のパケットタイプを処理できます。

- NULL FLIT
- PRET (シングル FLIT パケット)
- TRET (シングル FLIT パケット)
- IRTRY (シングル FLIT パケット)
- ERROR 応答 (シングル FLIT パケット)
- WRITE 応答 (シングル FLIT パケット)
- 16 バイトの READ 応答 (2-FLIT パケット)
- 32 バイトの READ 応答 (3-FLIT パケット)
- 48 バイトの READ 応答 (4-FLIT パケット)
- 64 バイトの READ 応答 (5-FLIT パケット)
- 80 バイトの READ 応答 (6-FLIT パケット)
- 96 バイトの READ 応答 (7-FLIT パケット)
- 112 バイトの READ 応答 (8-FLIT パケット)
- 128 バイトの READ 応答 (9-FLIT パケット)
- MODE READ 応答 (2-FLIT パケット)
- MODE WRITE 応答 (シングル FLIT パケット)

HMC コントローラー IP コアは、特定のベンダー固有のパケットタイプの定義やサポートをしません。

1.3 デバイスファミリー・サポート

次の表は、インテル FPGA IP コアのデバイス・サポート・レベルの定義をリストしています。

表 1. インテル FPGA IP コアのデバイス・サポート・レベル

FPGA デバイスファミリー
アドバンスサポート — IP コアは、このデバイスファミリーのシミュレーションおよびコンパイルに使用できます。FPGA プログラミング・ファイル (.pof) のサポートは Quartus® Prime Pro - Stratix 10 Edition Beta ソフトウェアではサポートされず、また、IP タイミング・クロージャータは保証されません。タイミングモデルには、初期のレイアウト後の情報に基づいた遅延の初期の工学的な見積りが含まれています。シリコンテストが実際のシリコンとタイミングモデルの相関を改善するために、タイミングモデルは変更される可能性があります。この IP コアは、システム・アーキテクチャーおよびリソース使用率の調査、シミュレーション、ピン配置、システム・レイテンシー評価、基本タイミング・アセスメント (パイプライン・バジェット)、および I/O 転送ストラテジー (データ幅、バースト深度、I/O 規格のトレードオフ) に使用できます。
暫定サポート — IP コアは、このデバイスファミリーの暫定タイミングモデルで検証されます。IP コアはデバイスファミリーの機能要件をすべて満たしていますが、タイミング解析は実施中の場合があります。量産デザインでの使用には注意が必要です。
最終サポート — IP コアは、このデバイスファミリーの最終タイミングモデルで検証されます。IP コアはデバイスファミリーの機能要件とタイミング要件をすべて満たしており、量産デザインで使用可能です。

次の表は、各インテルのデバイスファミリーにおける HMC コントローラー IP コアで提供されるサポートのレベルを示しています。

表 2. Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェアの HMC コントローラー IP コア・デバイスファミリーのサポート

デバイスファミリー	サポート
Stratix 10	アドバンス
他のすべてのデバイスファミリー	このリリースでは未サポート

関連情報

タイミングおよび電力モデル

Quartus Prime プロ・エディション・ソフトウェアの現バージョンでのデフォルトのデバイス・サポート・レベルを報告します。

1.4 IP コアの検証

インテル は、HMC コントローラー IP コアのバージョンをリリースする前に、現在のバージョンの QuartusPrime ソフトウェアで総合的なレグレッション・テストを実行します。HMC コントローラー IP コアは、機能を確認するためにシミュレーションおよびハードウェアでテストされます。

注意: Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェアでは、IP コアは、シミュレーションおよびコンパイルのみでテストされます。

関連情報

- [Knowledge Base Errata for HMC Controller IP core](#)
機能上の正確さの例外は、HMC コントローラー IP コアエラッタに記載されています。
- [インテル IP Release Notes](#)
HMC コントローラー IP コアの変更は、Quartus II ソフトウェア v15.0 から始まるインテル FPGA IP Release Notes に記載されています。インテル FPGA IP Release Notes のリストは主要な製品リリースで変更されています。

1.4.1 シミュレーション

インテル は Micron HMC BFM を使用して HMC コントローラー IP コアで次のテストを実行します。

- ランダムなリーガル・ペイロード・サイズおよびコンテンツをカバーする制約付きランダムテスト
- 仕様に対する IP コアの適切な動作の確認のためのアサーション・ベースのテスト
- 広範囲に及ぶパケットリトライ機能

制約付きランダム技法は、IP コアの機能検証に適切な刺激を生成します。インテルは、すべての重要な機能が検証されていることを確認するために、ライン、表現およびアサーションのカバレッジ測定基準を監視します。



1.5 性能およびリソース使用率

表 3. HMC コントローラー IP コアにおける FPGA リソース使用率

表は、次の IP コア機能をオフにし、ソフトウェアを使用した 10 Gbps のデータレートでコンフィグレーションされた HMC コントローラー IP コアの一般的なリソース使用率です。

- M20K ECC サポート

ALM とロジックレジスターの数は 100 に切り上げられます。丸め処理前の ALM 数は、Quartus Prime Pro – Stratix 10 Edition Beta Fitter Report からの必要な ALM 数です。

Stratix 10 IP コア・バリエーション		リソース使用率		
応答リオーダーリング	ポート数	必要な ALM 数	専用ロジックレジスター数	M20K ブロック数
オフ	1	26500	51000	51
	2	31700	62600	86
	3	36800	74300	121
	4	42000	86100	154
オン	1	32000	62000	55
	2	39800	80400	94
	3	49700	100900	133
	4	55300	116900	170

関連情報

- [Fitter Resources Reports in the Quartus Prime Pro Edition Help](#)
必要な ALM 数を含んだ Quartus Prime プロ・エディションのリソース使用率レポートに関する情報を提供します。
- [Quartus Prime Standard Edition Handbook Volume 1: Verification](#)

1.6 デバイスのスピードグレードのサポート

表 4. 最小推奨デバイスファミリーのスピードグレード

インテルは HMC コントローラー IP コアは、表にリストされているデバイスのスピードグレード、または使用可能な高速（より低い番号）デバイスのスピードグレードでのみコンフィグレーションすることを推奨します。

インテルは、低速（より高い番号）デバイスのスピードグレードでのこの IP コアのコンフィグレーションをサポートしていません。

デバイスファミリー	IP コア・バリエーション : レーンレート		
	10 Gbps	12.5 Gbps	15 Gbps
Stratix 10	(1)	(1)	(1)

(1) この情報は現在使用できません。



1.7 リリース情報

表 5. HMC コントローラー IP コアの現在のリリース情報

項目	バリュー
バージョン	Quartus Prime Pro - Stratix 10 Edition Beta
リリース日	2016.08.08
製品コード	IP-HMCSR15FW
ベンダー ID	6AF7



2 HMC コントローラー IP コアのスタート

ハイブリッド・メモリー・キューブ・コントローラー IP コアのインストール、パラメーター化、およびシミュレーションの方法について、以下の項で説明します。

11 ページの [IP コアのライセンス取得](#)

HMC コントローラー IP コアは Altera IP Library にある Quartus Prime ソフトウェアで使用可能です。

12 ページの [IP コアのパラメーターとオプションの指定](#)

HMC コントローラー IP コアは、標準のカスタマイズおよび生成プロセスをサポートしています。この IP コアは Qsys ではサポートされません。

13 ページの [HMC コントローラー IP コアのパラメーター](#)

18 ページの [インテル FPGA IP コア向けに生成されるファイル](#)

19 ページの [デザインへの IP コアの統合](#)

25 ページの [インテル FPGA IP コアのシミュレーション](#)

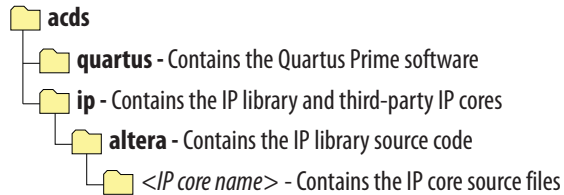
関連情報

- [60 ページの HMC コントローラー IP コア Stratix 10 のデザイン例](#)
HMC コントローラーのデザイン例は、IP コアを外部 I²C マスターモジュールと外部 TX PLL に接続する方法を提供します。
- [インテル FPGA IP コアについて](#)
すべてのインテル FPGA IP コアについて、パラメーター化、生成、アップグレード、および IP コアのシミュレーションといった基本的な情報を提供します。
- [Creating Version-Independent IP and Qsys Simulation Scripts](#)
ソフトウェアあるいは IP のバージョンのアップグレードのためのマニュアルでの更新を必要としないシミュレーション・スクリプトの作成について詳しい情報を提供します。
- [Project Management Best Practices](#)
プロジェクトおよび IP ファイルの効果的な管理および移植性のためのガイドラインを提供します。

2.1 IP コアのライセンス取得

インテル IP ライブラリーは、追加のライセンスなしでユーザーの製品に使用できる、役立つ IP コア機能を数多く提供しています。一部の MegaCore[®] IP ファンクションでは、製品に使用する場合に個別のライセンスの購入が必要になります。ただし、OpenCore[®] 機能を使用することにより、どの IP コアも Quartus Prime ソフトウェアでシミュレーションとコンパイルを行い評価することが可能です。機能性と性能にご満足いただいてから、Self Service Licensing Center ページで任意の MegaCore 製品のライセンス番号を取得いただけます。

図 -2: IP コアのインストールパス



注意: Windows でのデフォルトの IP インストール・ディレクトリーは `<drive>:\altera\<version number>` です。Linux の IP インストール・ディレクトリーは `<home directory> /altera/ <version number>` です。

2.1.1 OpenCore Plus IP 評価機能

アルテラの無償の OpenCore Plus 機能により、ライセンスが必要な MegaCore IP コアを購入前にシミュレーションおよびハードウェアで評価していただけます。MegaCore IP コアのライセンスの購入は、デザインの製品化を決定するまで必要ではありません。OpenCore Plus を使用すると、次の評価が行えます。

- システム内でのライセンスが必要な IP コアの動作のシミュレーション
- IP コアの機能性、サイズ、速度の迅速かつ容易な検証
- IP コアを含むデザイン向けの時間制限付きデバイス・プログラミング・ファイルの生成
- ハードウェア上での IP コアを含めたデバイスのプログラミングおよびデザインの検証

OpenCore Plus 評価機能は、次の 2 種類の動作モードをサポートします。

- Untethered (アンテザード)—ライセンスが必要な IP を含んだデザインを時間制限内で実行する。
- Tethered (テザード)—ライセンスが必要な IP を含んだデザインをさらに長時間、または無制限に実行する。この動作はボードとホスト・コンピュータが接続されている必要がある。

注意: デザイン内のいずれかの IP コアがタイムアウトすると、OpenCore Plus を使用するすべての IP コアが同時にタイムアウトします。

関連情報

- [Quartus Prime Licensing Site](#)
- [Quartus Prime Installation and Licensing](#)

2.2 IP コアのパラメーターとオプションの指定

HMC コントローラーの Parameter Editor を使用すると、カスタム IP バリエーションを素早く設定できます。Quartus Prime ソフトウェアで IP コアのオプションとパラメーターを指定するには、次の手順を実行します。



1. IP Catalog (**Tools > IP Catalog**) の **Memory Interfaces and Controllers** で、カスタマイズする IP コアの名前を見つけ、ダブルクリックします。Parameter Editor が表示されます。
2. カスタム IP バリエーションのトップレベルの名前を指定します。Parameter Editor は IP バリエーションの設定を Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェアの `<your_ip>.ip` と名付けられたファイル名に保存します。**OK** をクリックします。
3. Parameter Editor で、IP バリエーションに対してパラメーターと 1 つあるいは複数のオプションを指定します。特定の IP コア・パラメーターについては、Parameters セクションを参照してください。
4. **Generate HDL** をクリックします。**Generation** ダイアログボックスが表示されます。
5. HMC コントローラー IP コアのシミュレーション・モデルを生成するには、**Simulation > Create Simulation Model** にある **Verilog HDL** を選択してください。
6. その他の出力ファイル生成オプションを指定し、**Generate** をクリックします。IP バリエーション・ファイルは、仕様に従って生成されます。
7. **Finish** をクリックします。Parameter Editor が現在のプロジェクトにトップレベルの `.qsys` ファイルを自動で追加します。`.qsys` ファイルをプロジェクトに手動で追加するには、**Project > Add/Remove Files in Project** の順でクリックします。
8. IP バリエーションを生成してインスタンス化した後、ポート接続用の適切なピン割り当てを行います。

2.3 HMC コントローラー IP コアのパラメーター

HMC コントローラーの Parameter Editor は HMC コントローラー IP コアおよびシミュレーション・テストベンチをコンフィグレーションできるパラメーターを提供します。

HMC コントローラーの Parameter Editor は **Example Design** タブを含んでいます。タブについて詳しくは、[Hybrid Memory Controller Design Example User Guide](#) を参照してください。

表 6. HMC コントローラー IP コアのパラメーター

HMC コントローラー Parameter Editor の **IP** タブで HMC コントローラー IP コアをカスタマイズするためのパラメーターのリストです。

パラメーター	タイプ	範囲	デフォルト設定	パラメーターの説明
Lanes	整数	<ul style="list-style-type: none"> • 16 	16	選択全幅 (16 レーン) の機能を選択します。Stratix 10 デバイスをターゲットとする IP コアは、半幅の機能をサポートしていません。
Data rate	文字列	<ul style="list-style-type: none"> • 10 Gbps • 12.5 Gbps • 15 Gbps 	10 Gbps	各レーンでデータレートを選択します。
CDR reference clock	文字列	<ul style="list-style-type: none"> • 125 MHz • 156.25 MHz • 166.67 MHz (10 Gbps または 15 Gbps の Data rate でのみ使用可能) 	125 MHz	RX CDR PLL での入力リファレンス・クロックの周波数を選択します。このパラメーターで指定する周波数で <code>rx_cdr_refclk0</code> 入力信号を駆動する必要があります。また、このクロック、外部トランシーバー TX PLL リファレンス・クロック、および外部 HMC デバイスの <code>REFCLKP</code> および <code>REFCLKN</code> 入力信号を同じクロックソースから派生させる必要があります。

continued...



パラメーター	タイプ	範囲	デフォルト設定	パラメーターの説明
Ports	整数	<ul style="list-style-type: none"> • 1 • 2 • 3 • 4 	1	<p>ポート (データバス・インターフェイス) の数です。</p> <p>ポート数を増やすと、ハイブリッド・メモリー・キューブの使用率が向上し、効率が向上します。ただし、着信要求の中で IP コアがアービトレーションするため、個々のポートで request-to-response レイテンシーが増加する可能性があります。</p> <p>複数のポートを指定すると、各ポートには一連のタグが割り当てられます。</p> <ul style="list-style-type: none"> • 2 つのポートを指定する場合、ポート 0 は 0 ~ 255 の範囲、ポート 1 は 256 ~ 511 の範囲のタグを、それぞれ使用する必要があります。 • 3 つのポートを指定する場合、ポート 0 は 0 ~ 175 の範囲、ポート 1 は 176 ~ 351 の範囲、ポート 2 は 352 ~ 511 の範囲のタグを、それぞれ使用する必要があります。 • 4 ポートを指定する場合、ポート 0 は 0 ~ 127 の範囲、ポート 1 は 128 ~ 255 の範囲、ポート 2 は 256 ~ 383 の範囲、ポート 3 は 384 ~ 511 の範囲のタグを、それぞれ使用する必要があります。
Enable response re-ordering	ブーリアン	<ul style="list-style-type: none"> • True • False 	False	<p>応答が元の要求が対応する要求インターフェイスに到着した順序に各データ応答インターフェイスに表示されることを IP コアが確認するかどうかを指定します。</p> <p>この機能をオンにすると、IP コアはタグを内部で管理します。その場合、タグはデータ・インターフェイス上で使用できません。</p> <p>この機能をオンにすると、round-trip レイテンシーが増加する可能性があります。</p>
RX mapping	64 ビット値		0xFEDCBA9876543210	<p>RX レーン・マッピングを選択します。</p> <p>このパラメーターの変更には注意が必要です。RX Mapping パラメーターおよび TX Mapping パラメーター 15 ページの を参照してください。</p>
TX mapping	64 ビット値		0xFEDCBA9876543210	<p>TX レーンマッピングを選択します。</p> <p>このパラメーターの変更には注意が必要です。RX Mapping パラメーターおよび TX Mapping パラメーター 15 ページの を参照してください。</p>
Enable M20K ECC support	Boolean	<ul style="list-style-type: none"> • True • False 	False	<p>IP コアが、IP コアの一部として構成されたデバイス M20K メモリーブロックの ECC 機能をサポートするかどうかを指定します。</p> <p>このパラメーターをオンにすると、IP コアで構成された M20K メモリーブロック内の 1 ビット・エラー訂正、隣接する 2 ビット・エラー訂正、および 隣接する 3 ビット・エラー訂正 ECC 機能を有効にして、データの信頼性が高められます。このパラメーターをオフにすると、レイテンシーとリソース使用率が減少します。</p>



2.3.1 RX Mapping パラメーターおよび TX Mapping パラメーター

HMC コントローラー IP コアは、柔軟なボード設計のために **RX mapping** パラメーターおよび **TX mapping** パラメーターを提供しています。

これらのパラメーターのデフォルト値は、HMC デバイスの LxTX[<i> </i>] 出力信号を HMC コントローラー IP コアの hmc_lxrx[<i> </i>] 入力ポートに接続し、LxRX[<i> </i>] 入力信号を hmc_lxtx[<i> </i>] 出力ポートに接続する際に、それぞれの <i> </i> で、正しい IP コアの動作を指定します。

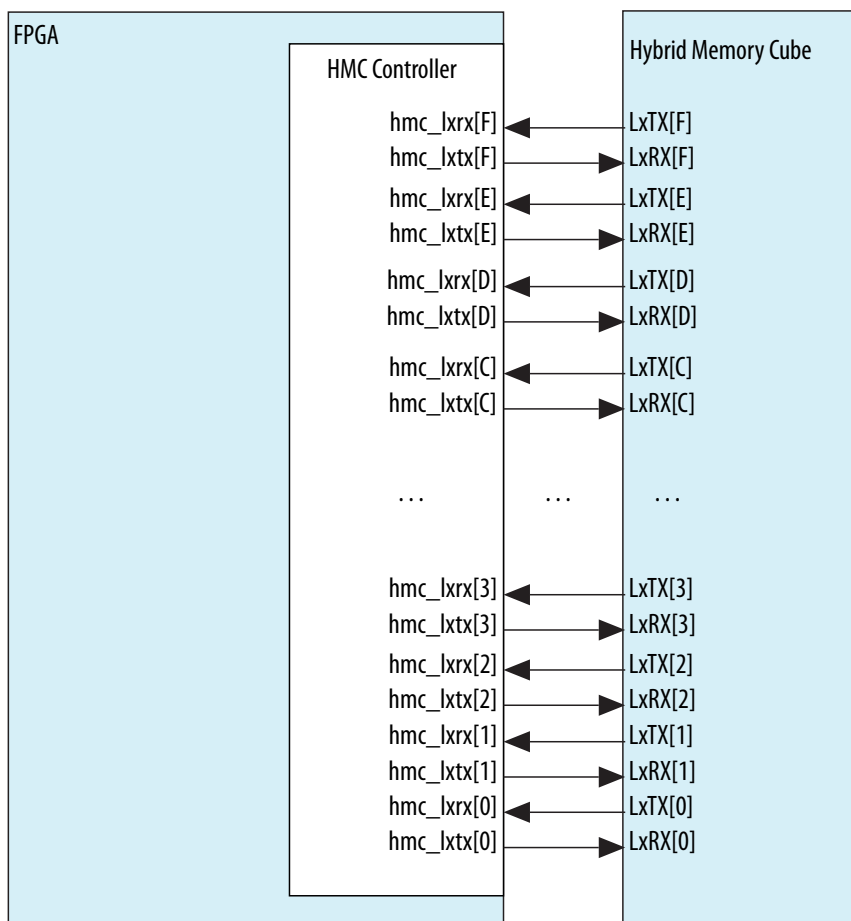
ただし、デザイン上の制約により、これらの信号が予期どおりに接続されない場合は、HMC コントローラー IP コアの mapping パラメーターの一方または両方を標準ではない接続に対応させるために変更できます。

注意: Quartus Prime Fitter を使用すると、HMC コントローラー IP コアレーンデバイストランシーバーのチャンネルに順にマッピングができなくなります。したがって、これらの 2 つのパラメーターは、デバイストランシーバーピンと HMC デバイSPORTの間のボード上の順序外の接続を補うのみです。

図 -3: デフォルトの RX と TX Mapping パラメーター値

RX mapping value 0xFEDCBA9876543210

TX mapping value 0xFEDCBA9876543210





HMC デバイスの LxTX[<i>] 出力信号を HMC コントローラーの IP コアの hmc_lxrx[<k>] 入力ポートに接続する場合、**RX mapping** パラメーターのビット [(4<i>+3):(4<i>)] (ニブル <i>) の値を 4'h<k> に設定します。したがって、**RX mapping** パラメーターのデフォルト値は 0xFEDCBA9876543210 であり、LxTX[F] は hmc_lxrx[F] に接続し、LxTX[E] は hmc_lxrx[E] に接続することを表示します。

HMC デバイスの LxRX[<i>] 入力信号を HMC コントローラー IP コアの hmc_lxtx[<k>] 入力ポートに接続する場合、**TX mapping** パラメーターのビット [(4<i>+3):(4<i>)] (ニブル <i>) の値を 4'h<k> に設定します。したがって、**TX mapping** パラメーターのデフォルト値は 0xFEDCBA9876543210 であり、LxRX[F] は hmc_lxtx[F] に接続し、LxRX[E] は hmc_lxtx[E] に接続することを表示します。

例 : デフォルト以外の RX Mapping パラメーター値

表 7. デフォルト以外の RX 接続

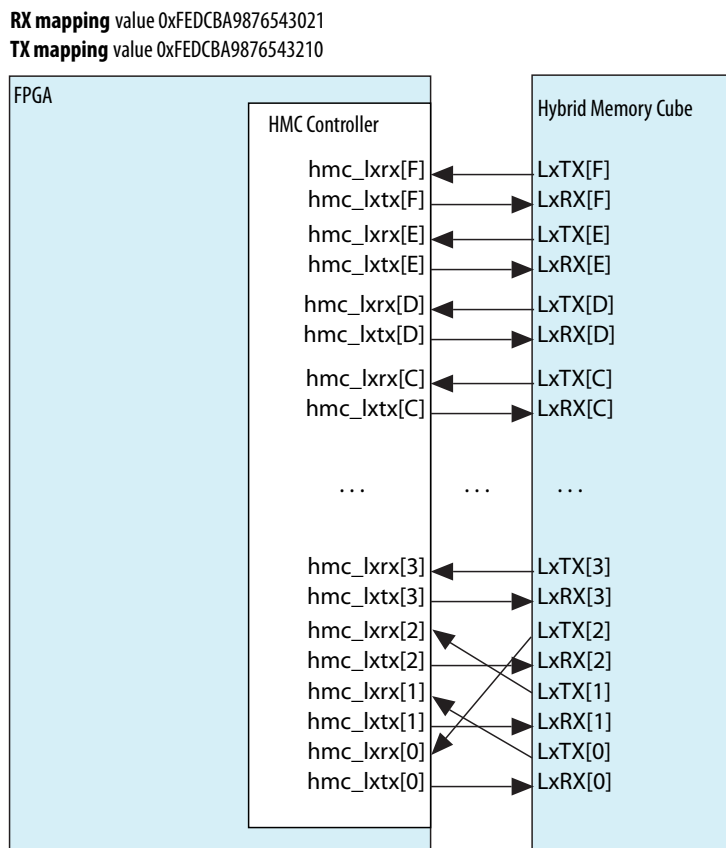
HMC デバイス出力信号	IP コア入力信号
LxTX[2]	hmc_lxrx[0]
LxTX[1]	hmc_lxrx[2]
LxTX[0]	hmc_lxrx[1]



図 -4: デフォルト以外の RX Mapping パラメーター値の例

HMC コントローラー IP コアの hmc_lxrx[2:0] 入力信号を表に従って接続し、その他すべての IP コアの hmc_lxrx[<i>i</i>] 入力ポートを対応する HMC デバイスの LxTX[<i>i</i>] 出力ポートに接続する場合、標準ではない接続を補償するために **RX mapping** パラメーターの値を 0xFEDCBA9876543021 に設定します。

注意: **RX mapping** パラメーターは位置と値による IP コアレーンにより、HMC デバイスレーンを指定します。図は 0xFED.....43021 の mapping パラメーター値を表し、0xFED....43102 の値ではありません。



例 : デフォルト以外の TX Mapping パラメーター値

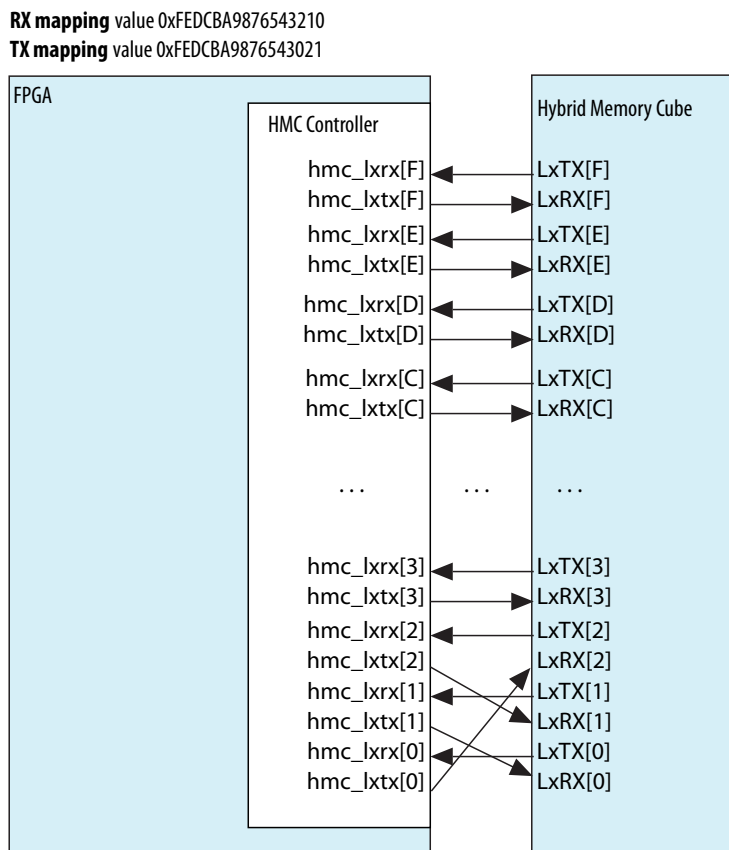
表 8. デフォルト以外の TX 接続

HMC デバイス入力信号	IP コア出力信号
LxRX[2]	hmc_lxtx[0]
LxRX[1]	hmc_lxtx[2]
LxRX[0]	hmc_lxtx[1]

図 -5: デフォルト以外の TX Mapping パラメーター値の例

HMC コントローラー IP コアの `hmc_lxtx[2:0]` 出力信号を表に従って接続し、その他すべての IP コアの `hmc_lxtx[<i></i>]` 出力ポートを対応する HMC デバイスの `LxRX[<i></i>]` 入力ポートに接続する場合、非標準接続を補償するために **TX mapping** パラメーターの値を `0xFEDCBA9876543021` に設定します。

注意: **TX mapping** パラメーターは位置と値による IP コアレーンにより、HMC デバイスレーンを指定します。図は `0xFED.....43021` の `m apping` パラメーター値を表し、`0xFED....43102` の値ではありません。

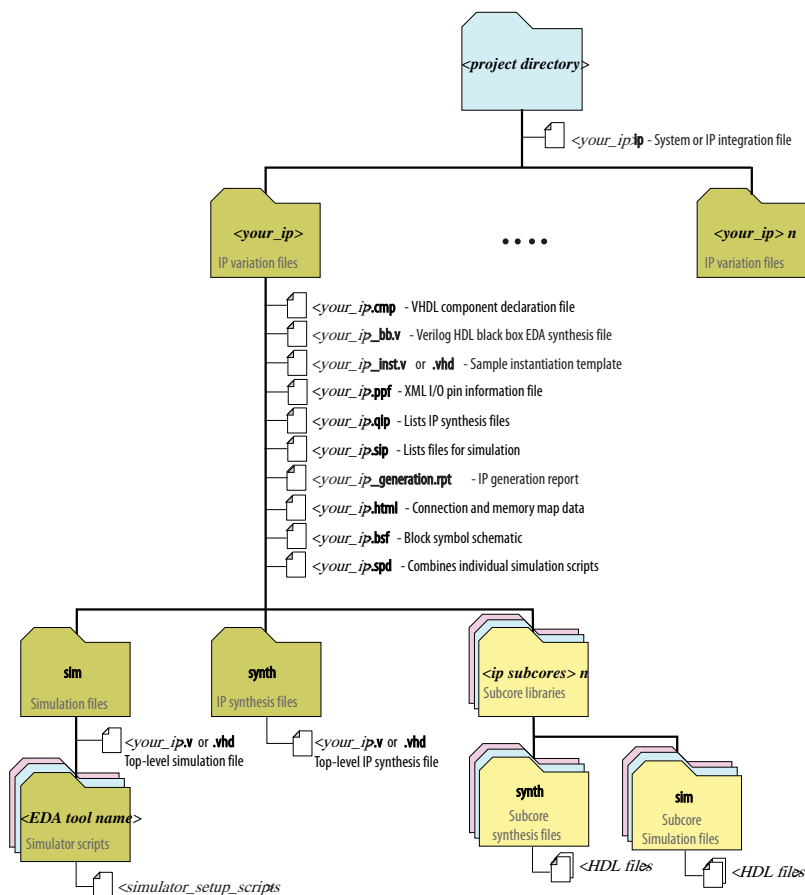


これらのパラメーターの変更には注意が必要です。ループバック設定では、**RX mapping** と **RX mapping** パラメーターが逆のマッピングを指定するようする必要があります。そうでない場合、RX レーンスワッパーのダウンストリームにある IP コアは、誤ったレーンでデータを受信するために現れません。

2.4 インテル FPGA IP コア向けに生成されるファイル

Quartus Prime ソフトウェアは IP コア・バリエーションの生成中に複数のファイルを生成します。

図 -6: IP コアが生成するファイル



2.5 デザインへの IP コアの統合

HMC コントローラー IP コアがハードウェアで正しく機能するようにするには、追加のブロックを IP コアに接続し、デバイスピンを順に割り当てる必要があります。

2.5.1 ピンの制約

HMC コントローラー IP コアのインスタンスをデザインに統合する際は、適切なピン割り当てを行う必要があります。シミュレーションやハードウェアにデザインをマップする準備ができていない間は、トップレベルの信号に特定のピン割り当ての作成を回避するために、仮想ピンを作成することができます。

デザインをハードウェアにマップする準備ができたなら、次の制約を適用する必要があります。

- 隣接する HMC コントローラーのレーンは、隣接するインテルのデバイスのピンにマップする必要があります。他のインテルのデバイスピンへのレーンのマッピングによるレーン順序の入れ替えはできません。代わりに、**RX mapping** および **TX mapping** パラメーターを使用して、ボード設計の問題を補えます。
- Stratix 10 デバイスをターゲットとする HMC コントローラー IP コアのすべてのレーンは、同じトランシーバー・タイルでコンフィグレーションする必要があります。
- HMC コントローラー IP コアのレーンは、3 つ以下のトランシーバー・ブロックで構成する必要があります。この制約を適用するには、次の制限でトランシーバー・チャンネルの IP コアレーンを設定する必要があります。
 - レーン 0 はトランシーバー・ブロックのチャンネル 0、1 または 2 にマップする必要があります。
 - レーン 0 をチャンネル 0 にマップする場合、HMC コントローラー・レーン 1 は同じトランシーバー・ブロック (トランシーバー・ブロック N) のチャンネル 1 にマップし、レーン 15 をトランシーバー・ブロック N+2 のチャンネル 3 にマップする必要があります。
 - レーン 0 をチャンネル 1 にマップする場合、HMC コントローラー・レーン 1 は同じトランシーバー・ブロック (トランシーバー・ブロック N) のチャンネル 2 にマップし、レーン 15 をトランシーバー・ブロック N+2 のチャンネル 4 にマップする必要があります。
 - レーン 0 をチャンネル 2 にマップする場合、HMC コントローラー・レーン 1 を同じトランシーバー・ブロック (トランシーバー・ブロック N) のチャンネル 3 にマップし、レーン 15 をトランシーバー・ブロック N+2 のチャンネル 5 にマップする必要があります。

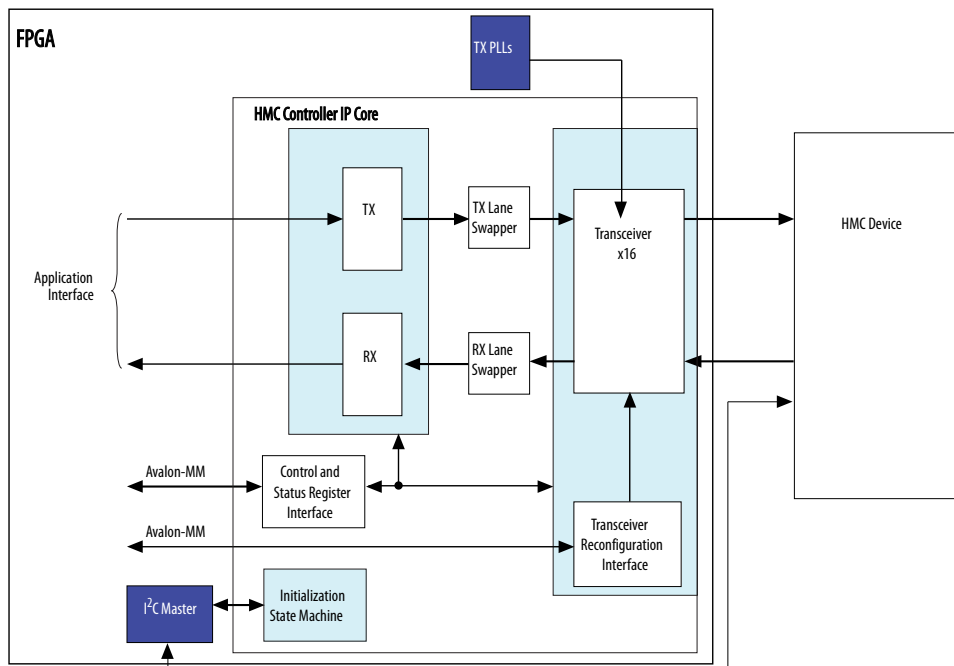
2.5.2 必要な外部インターフェイス

HMC コントローラー IP コアは、次の追加モジュールを定義してインスタンス化する必要があります。

- すべての HMC レーンでトランシーバー TX PLL をコンフィグレーションするための外部 PLL IP コア—これらの IP コアが構成するハードウェアは物理的にはデバイス・トランシーバーの一部である可能性があります。HMC コントローラー IP コアとは別にソフトウェアでこれらをインスタンス化する必要があります。この要件は、デバイス内の同じトランシーバー・ブロックを使用する複数のインテル FPGA IP コアのコンフィグレーションをサポートします。
- デザイン上の外部 I²C マスターモジュール— IP コアが接続する HMC デバイスを初期化するには、このモジュールを含める必要があります。

図 -7: 必要な外部ブロック

必要な外部ブロックは、図では他のブロックより濃い色で表示されます。外部 TX PLL IP コアは、デバイス・トランシーバーの ATX PLL またはトランシーバー・モードでの fPLL を構成します。



2.5.2.1 外部 PLL の追加

HMC コントローラー IP コアは、外部 PLL IP コアを生成し、接続する必要があります。HMC コントローラー IP コアレーンとして構成された TX トランシーバー・チャンネルをクロックするために必要な PLL IP コアを生成する必要があります。ATX PLL IP コアは、トランシーバー内のトランシーバー PLL をハードウェアで構成しますが、HMC コントローラー IP コアとは別にソフトウェアでトランシーバー PLL IP コアを生成する必要があります。あるいは、トランシーバー・モードで fPLL を構成することも可能です。PLL IP コアを生成して接続しない場合、HMC コントローラー IP コアはハードウェアで正しく機能しません。

IP Catalog を使用して、デバイスにトランシーバー PLL を構成する外部 PLL IP コアを生成することができます。

トランシーバー PLL パラメーター・エディターでは、xN ボンディング・コンフィグレーションで PLL IP コアを構成する必要があります。また、次のパラメーター値も設定する必要があります。

- **PLL output frequency** を IP コア・バリエーションの各レーンあたりのデータレートの半分に設定します。トランシーバーは、PLL からの入力クロックの立ち上がりエッジと立ち下がりエッジの両方を使用して、デュアル・エッジ・クロッキングを実行します。したがって、この PLL 出力周波数の設定は、HMC デバイスに接続するレーンの正しいクロックでトランシーバーを駆動します。
- **PMA interface width** を **32** に設定します。
- **PLL integer reference clock frequency** (ATX PLL) または **Desired reference clock frequency** (fPLL) を設定します。

インテルは 125 MHz、156.25 MHz、または 166.67 MHz の指定を推奨しています。理論的には、PLL が必要な出力クロック周波数を生成できる任意のリファレンス・クロック周波数を指定できます。ただし、この TX PLL と RX CDR PLL (HMC コントローラー IP コアへの rx_cdr_refclk0 入力信号) と HMC デバイスのリファレンス・クロック入力信号 (REFCLKP と REFCLKN) を同じクロックソースから駆動する必要があります。

注意: このパラメーターに指定する周波数で外部 PLL リファレンス・クロック入力信号を駆動する必要があります。

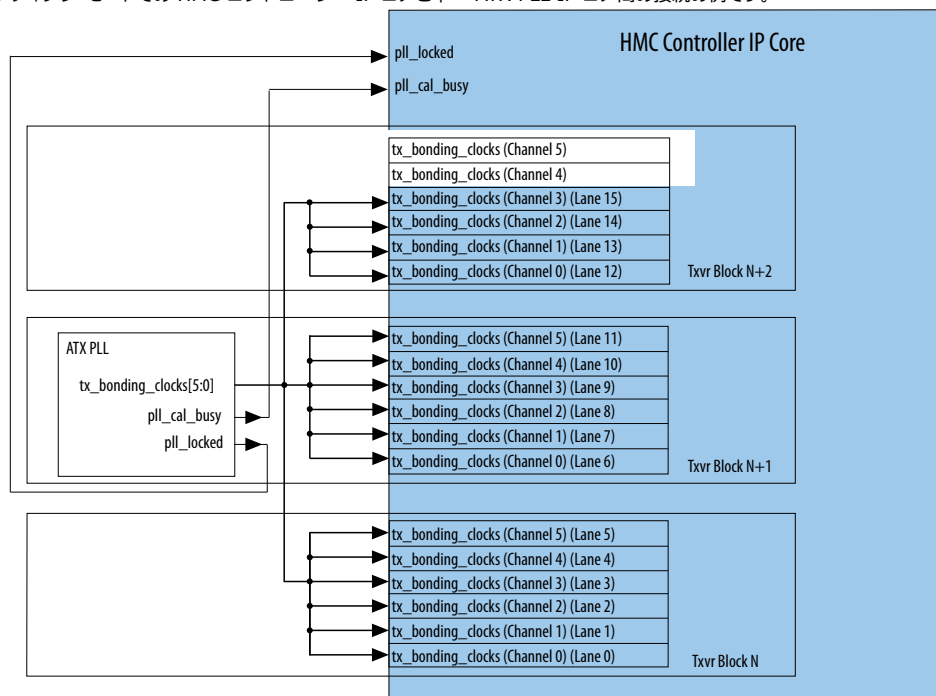
xN ボンディング・モードでは、構成されたトランシーバー・ブロックのチャンネルを駆動するには 1 つの PLL で十分です。隣接する物理トランシーバー・チャンネルで HMC リンク TX シリアルレーンを順に構成する必要があることを考慮し、これらのレーンが最大 3 つのトランシーバー・ブロックを構成するようにします。これらの要件を満たす I/O 制約は、HMC コントローラー IP コアに付属のデザイン例の Quartus Setting ファイルの hmcc_example.qsf で確認できます。

注意: HMC コントローラー IP コアは、PLL フィードバック補償のボンディングをサポートしていません。

PLL 出力はトランシーバー・ブロック用の x6 ネットワークに直接接続し、xN クロック・ネットワークを介して追加のトランシーバー・ブロックを駆動します。

図 -8: xN ボンディング手法によるトランシーバー PLL 接続の例

xN ボンディング・モードでの HMC コントローラー IP コアと単一 ATX PLL IP コア間の接続の例です。



外部 PLL 信号と HMC コントローラー IP コア・トランシーバーの TX PLL インターフェイス信号は、次の規則に従って接続する必要があります。

HMC コントローラー信号	TX PLL 信号に接続
HMC レーン N での tx_bonding_clocks[5:0] 入力信号	レーン N で構成されるトランシーバー・ブロックにおける PLL IP コアの tx_bonding_clocks[5:0] 出力ベクターです。 xN bonding の場合、単一 PLL は xN クロック・ネットワークに接続し、構成された PLL とは異なるトランシーバー・ブロックの HMC レーンの tx_bonding_clocks[5:0] 入力ピンが xN クロック・ネットワークからクロックを受信します。
pll_locked 入力信号	すべての HMC レーンの外部 PLL pll_locked 出力信号
pll_cal_busy 入力信号	すべての HMC レーンの外部 PLL の pll_cal_busy 出力信号

ユーザーロジックは接続を提供する必要があります。

関連情報

- 29 ページの [外部 PLL インターフェイス](#)
- 50 ページの [外部 PLL へのインターフェイス上の信号](#)
- 60 ページの [HMC コントローラー IP コア Stratix 10 のデザイン例](#)
HMC コントローラーのデザイン例は、外部 PLL を HMC コントローラー IP コアに接続する方法を提供します。
- 19 ページの [ピンの制約](#)
IP コアレーンが最大 3 つのトランシーバー・ブロックを構成する要件について説明します。

2.5.2.2 外部 I²C マスターモデルの追加

HMC コントローラー IP コアは、デザインで外部 I²C をマスターモジュールをインスタンス化する必要があります。デザインはこのモジュールを含めて IP コアが接続する HMC デバイスを初期化する必要があります。または、JTAG マスターモジュールを使用して必要な初期化シーケンスを実装することもできます。

システムの I²C マスターモジュールは、システム内の特定の HMC デバイスの初期化要件に従って、HMC デバイス・コンフィグレーション・レジスターをロードする必要があります。

HMC 仕様では、HMC を初期化するコマンドを送信した後、HMC デバイスの REGISTER REQUEST commands レジスターを Init Continue の値に設定する必要があります。したがって、I²C マスターモジュールは、HMC デバイス・コンフィグレーション・レジスターのロードシーケンスの正常終了を表示するために、このレジスターを設定する必要があります。

さらに、I²C マスターモジュールは、HMC コントローラー IP コアに接続するために、次の 2 つの信号を供給する必要があります。

- HMC デバイスのコンフィグレーション・レジスターをロードする要求を受信する入力信号。この信号を HMC コントローラー IP コアの `i2c_load_registers` 出力信号に接続する必要があります。複数の HMC コントローラー IP コアが同じ HMC デバイスに接続されている場合は、この入力信号を個別の HMC コントローラー IP コアの `i2c_load_registers` 出力信号の AND に接続する必要があります。AND 関数を提供する必要があります。
- コンフィグレーション・レジスターのロードシーケンスが正常に終了したことを示す出力信号。I²C マスターは、この信号を次の動作で実装する必要があります。
 1. リセット解除時にこの信号をディアサートします。
 2. Init Continue を HMC デバイスの REGISTER REQUEST commands レジスターに書き込んだ後、この信号をアサートします。
 3. 上記の入力信号の立ち下がりエッジに応答して、この信号をディアサートします。

この信号を HMC コントローラー IP コアの `i2c_registers_loaded` 入力信号に接続する必要があります。複数の HMC コントローラー IP コアが同じ HMC デバイスに接続されている場合、この信号をすべての HMC コントローラー IP コアの `i2c_registers_loaded` 信号に接続する必要があります。

必要なレジスター・コンフィグレーション・シーケンスについては、HMC コントローラー IP コアに接続されている HMC デバイスのデータシートを参照する必要があります。HMC コントローラー IP コアは Response Open Loop モードで動作することを考慮し、IP コアと正しく通信するようにこのモードで HMC デバイスをコンフィグレーションする必要があります。さらに、IP コアは TGA フィールドをサポートしていないため、ライト応答パケットですべてのノンポストド Write 要求に応答するように HMC デバイスを構成する必要があります。

関連情報

- 60 ページの [HMC コントローラー IP コア Stratix 10 のデザイン例](#)
HMC コントローラーのデザイン例は、外部 I²C マスターモジュール、およびそれを HMC コントローラー IP コアに接続する方法を提供します。
- 28 ページの [外部 I²C マスターへのインターフェイス](#)
- 46 ページの [I²C マスターへのインターフェイス上の信号](#)
このインターフェイス上の信号、HMC コントローラー IP コアが実装する 4 通りのハンドシェーキング・プロトコル、および I²C マスターが正しい IP コア機能を実装する必要があることについて説明します。

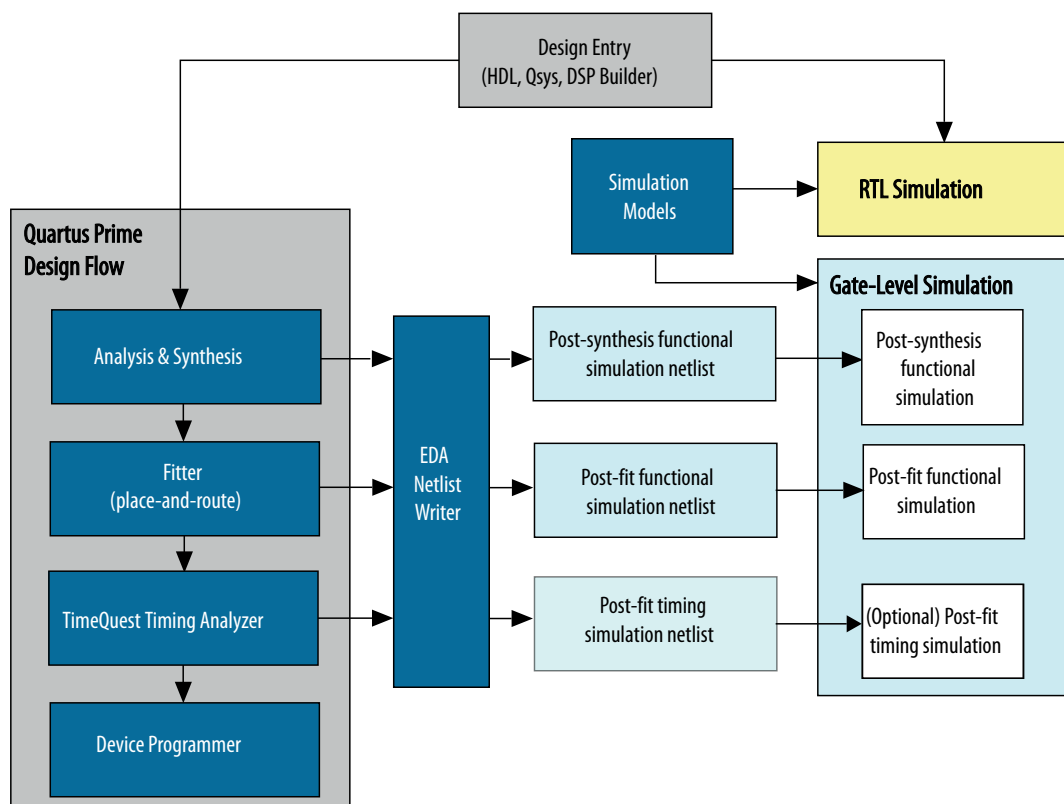
- **Hybrid Memory Cube Specification 1.1**
初期化シーケンスの要件については、Hybrid Memory Cube 仕様の *Power-On and Initialization* の項に記載されています。

2.6 インテル FPGA IP コアのシミュレーション

Quartus Prime ソフトウェアは、サポートされる EDA シミュレーションでインテル FPGA IP コアの RTL およびゲートレベルのデザイン・シミュレーションをサポートしています。シミュレーションには、シミュレーターの動作環境のセットアップ、シミュレーション・モデル・ライブラリーのコンパイル、およびシミュレーション実行が含まれます。

シミュレーションには、IP コアで使用可能な機能シミュレーション・モデルとテストベンチまたはデザイン例を使用することができます。**Generate Example Design** ボタンをクリックすると、指定した場所に機能シミュレーション・モデルとテストベンチ・ファイルが生成されます。デフォルトでは、ターゲットの場所を変更しない場合はプロジェクトのサブディレクトリーに生成されます。このディレクトリーには、テストベンチをコンパイルし、実行するためのスクリプトが含まれています。IP コアのシミュレーションに必要なモデルの完全なリストまたはライブラリーについては、テストベンチで生成されるスクリプトを参照してください。

図 -9: Quartus Prime デザインフローのシミュレーション



注意: ポストフィット・タイミング・シミュレーションは 28nm 以降のデバイスのアーキテクチャーではサポートされません。したがって、HMC コントローラー IP コアはポストフィット・タイミング・シミュレーションをサポートしません。



インテル FPGA IP は、シミュレーション特有の IP 機能シミュレーション・モデルや暗号化された RTL モデル、プレーンテキスト RTL モデルなどのさまざまなシミュレーション・モデルをサポートしています。これらはすべてサイクル精度モデルです。モデルは、業界標準の VHDL または Verilog HDL シミュレーターを使用する IP コア・インスタンスの高速な機能シミュレーションをサポートしています。一部のコアでは、プレーンテキスト RTL モデルのみが生成され、そのモデルをシミュレーションできます。

注意: シミュレーション・モデルはシミュレーションのみに使用し、合成やその他の目的のために使用しないでください。これらのモデルを合成に使用すると、機能しないデザインが作成されます。

HMC BFM を使用して HMC コントローラー IP コアをシミュレーションする場合は、HMC コントローラー IP コアとデザインの機能に合わせて BFM パラメーターを設定してください。例えば、BFM メモリーサイズ (2G または 4G) をデザインがアクセスするアドレススペースと一致するように設定し、Response Open Loop モードで HMC コントローラー IP コアと正しく通信するように BFM を設定することを確認します。また、HMC コントローラー IP コアが TGA フィールドをサポートしていないため、受信されたノンポストド Write トランザクションのライト応答パケットを送信するように BFM を設定する必要があります。

関連情報

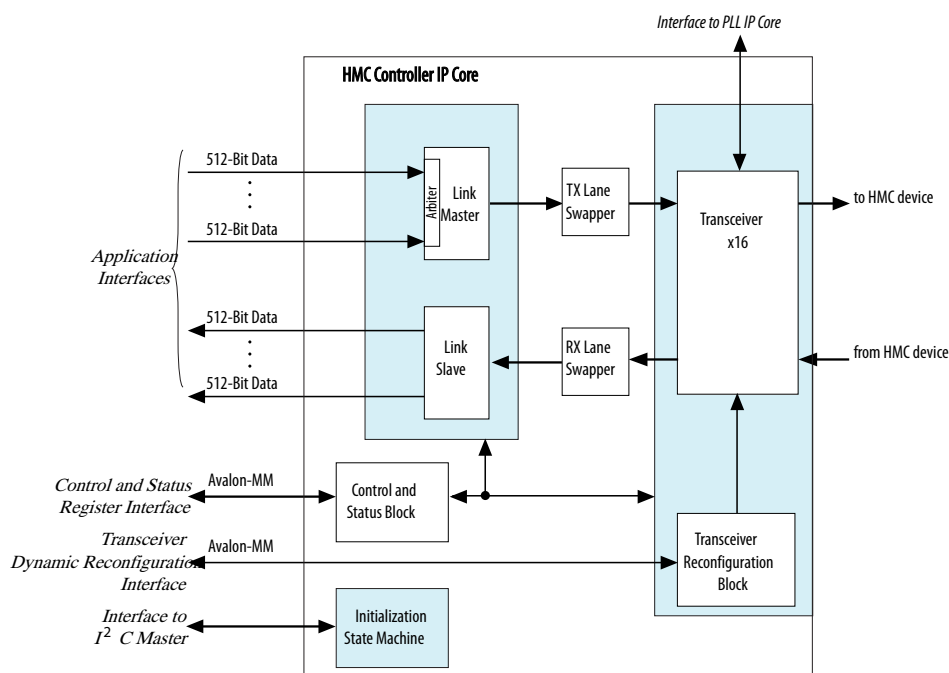
[Simulating Intel FPGA IP Cores](#)

3 機能概要

インテル HMC コントローラー IP コアは外部 HMC デバイスに容易にアクセスすることができます。

3.1 上位レベルのブロック図

図 -10: HMC コントローラー IP コアのブロック図



HMC コントローラー IP コアは次のコンポーネントを含んでいます。

- HMC TX パスおよび HMC RX パスの 2 つのデータバス。各バスには、レイヤーモジュール、レーンスワッパー、および HMC リンク上の高速トランシーバーがあります。
- 初期化ステートマシン
- レジスター・コントロール・ブロック
- ネイティブ PHY ダイナミック・コンフィグレーション・ブロック。このブロックは、Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェアでは機能しません。

TX レーンスワッパーは **TX mapping** パラメーターに従って、HMC TX レーンをトランシーバー・チャンネルに再マッピングします。RX レーンスワッパーは、**RX mapping** パラメーターに従って HMC RX レーンをトランシーバー・チャンネルから再マッピングします。

3.2 インターフェイスの概要

HMC コントローラー IP コアは複数の外部インターフェイスをサポートしています。

3.2.1 アプリケーション・インターフェイス

データパス要求インターフェイスおよび応答インターフェイスはアプリケーション要求インターフェイスおよびアプリケーション応答インターフェイスとも呼ばれ、512 ビットのデータバスと、アプリケーションが HMC 要求パケットフィールド値を提供し、HMC 応答パケットフィールド値をリードするための専用信号を提供します。IP コアは、1 つ、2 つ、3 つ、または 4 つのペアのデータパス要求および応答インターフェイスをサポートします。

関連情報

37 ページの [アプリケーション・インターフェイス信号](#)

3.2.2 HMC インターフェイス

HMC インターフェイスは外部 HMC デバイスに接続し、HMC 仕様に準拠しています。このインターフェイスは、隣接する 16 個のトランシーバー・チャンネルのインテルのデバイスで構成された単一の 16 レーンリンクを提供します。

HMC コントローラー IP コアは Response Open Loop モードで動作します。

関連情報

44 ページの [HMC インターフェイス信号](#)

HMC コントローラー IP コアの HMC インターフェイスは、外部 HMC デバイスのリンク・インターフェイスおよびメインリセット信号に接続します。

3.2.3 外部 I²C マスターへのインターフェイス

HMC コントローラー IP コアはデザインの外部 I²C マスターモジュールまたは JTAG マスターモジュールのインスタンス化を必要とします。この外部マスターモジュールは、HMC と HMC コントローラーの間のリンク上のリンク初期化の調整を必要とします。マスターモジュールは HMC コントローラーの内部初期化ステートマシンと連携して IP コアが接続する HMC デバイスのコンフィギュレーション・レジスターをプログラムします。

HMC コントローラー IP コアと初期化マスターモジュールを分離することで、デザインの柔軟性が向上します。IP コアには I²C マスターモジュールが含まれていないため、複数の HMC コントローラー IP コアのリンク初期化を制御するために単一の I²C マスターをインスタンス化できます。外部 I²C マスターモジュールは他の I²C スレーブを制御することも可能です。

関連情報

- 24 ページの [外部 I2C マスターモデルの追加](#)
HMC コントローラー IP コアを外部 I2C マスターモジュールに接続する方法についての情報を提供します。
- 46 ページの [I2C マスターへのインターフェイス上の信号](#)
このインターフェイス上の信号、HMC コントローラー IP コアが実装する 4 通りのハンドシェーキング・プロトコル、および I²C マスターが正しい IP コア機能を実装する必要があることについて説明します。



3.2.4 コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイス

コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイスは、HMC コントローラー IP コアの内部コントロール・レジスターおよびステータスレジスターへのアクセスを提供します。このインターフェイスはトランシーバー・レジスターへのアクセスを提供しません。

コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイスは、*Avalon Interface Specifications* で定義されている Avalon メモリーマップド (Avalon-MM) 仕様に準拠しています。

コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイスは、レジスターコンテンツの 32 ビット幅のデータバスを提供します。すべての HMC コントローラーのコントロールおよびステータス・レジスター・インターフェイスは 32 ビット幅であり、コントロールおよびステータス・レジスター・インターフェイス経由のすべてのレジスターのアクセスは、32 ビットのレジスターコンテンツをリードまたはライトします。

関連情報

- 47 ページの [コントロール・レジスター・インターフェイス信号およびステータス・レジスター・インターフェイス信号](#)
- 51 ページの [HMC コントローラー IP コアのレジスターマップ](#)
- [Avalon Interface Specifications](#)

3.2.5 ステータス・インターフェイスおよびデバッグ・インターフェイス

ステータス・インターフェイスおよびデバッグ・インターフェイスは、正常なリンク初期化と通信し、HMC システムのデバッグをサポートするための信号を提供します。

関連情報

48 ページの [ステータス信号およびデバッグ信号](#)

3.2.6 トランシーバー・コントロール・インターフェイス

HMC コントローラー IP コアは、次のトランシーバー・コントロール・インターフェイスをサポートします。

29 ページの [外部 PLL インターフェイス](#)

30 ページの [トランシーバー・リコンフィギュレーション・インターフェイス](#)

3.2.6.1 外部 PLL インターフェイス

HMC コントローラー IP コアは、外部トランシーバー PLL IP コアを生成し、各 HMC コントローラー IP コアのレーンに接続する必要があります。

トランシーバー PLL IP コアを生成して接続しない場合、HMC コントローラー IP コアはハードウェアで正しく機能しません。

関連情報

- 21 ページの [外部 PLL の追加](#)
パラメーター要件を含めた外部トランシーバー PLL IP コアを生成する方法について説明します。

- 50 ページの [外部 PLL へのインターフェイス上の信号](#)
- 60 ページの [HMC コントローラー IP コア Stratix 10 のデザイン例](#)
HMC コントローラーのデザイン例は、外部 PLL を HMC コントローラー IP コアに接続する方法を提供します。

3.2.6.2 トランシーバー・リコンフィグレーション・インターフェイス

トランシーバー・リコンフィグレーション・インターフェイスはエンベデッド・ネイティブ PHY IP コアのレジスターへのアクセスを提供します。このインターフェイスは、デバイス上のハード PCS レジスターへの直接のアクセスを提供します。

注意: このインターフェイスは Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェアに含まれていますが、このリリースでは機能しません。ただし、IP コアは内部的に `reconfig_clk` を使用します。IP コアが正しく機能するようにするには、有効なクロックを `reconfig_clk` 入力ポートに接続する必要があります。

トランシーバー・リコンフィグレーション・インターフェイスは、*Avalon Interface Specifications* で定義されている Avalon メモリーマップド (Avalon-MM) 仕様に準拠しています。

関連情報

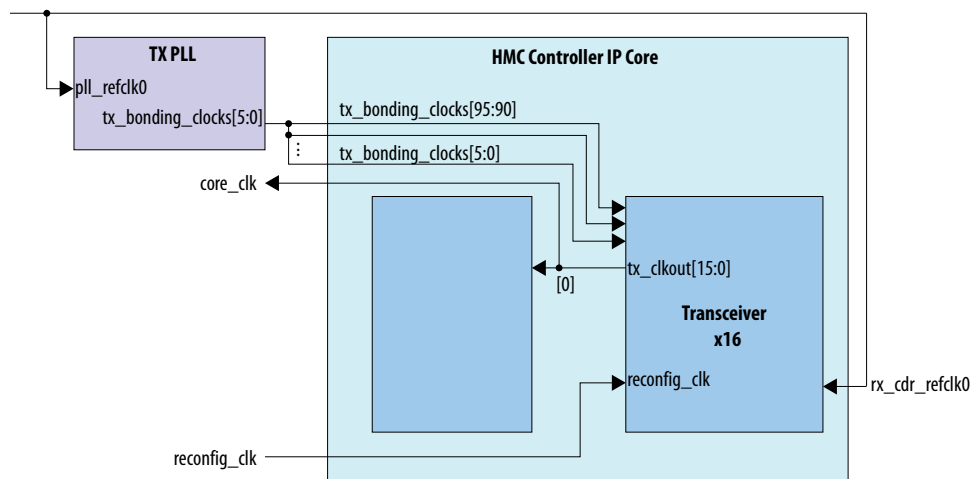
- 49 ページの [トランシーバー・リコンフィグレーション信号](#)
- [Avalon Interface Specifications](#)
Avalon Memory-Mapped (Avalon-MM) 仕様を定義します。

3.3 クロッキング構造

HMC コントローラー IP コアには、シングル・コア・クロックドメインと複数のトランシーバー関連のクロックドメインがあります。

デザインでは、外部トランシーバー TX PLL リファレンス・クロック、RX CDR リファレンス・クロック、および外部 HMC デバイスの `REFCLKP` および `REFCLKN` 入力信号を同じクロック・リファレンス・ソースから派生させる必要があります。この要件は、HMC 仕様で要求されるように、受信クロックと送信クロックの間に 0 PPM の差を保証します。

図 -11: HMC コントローラー IP コアのクロッキング図



関連情報

48 ページの [クロック信号およびリセット信号](#)

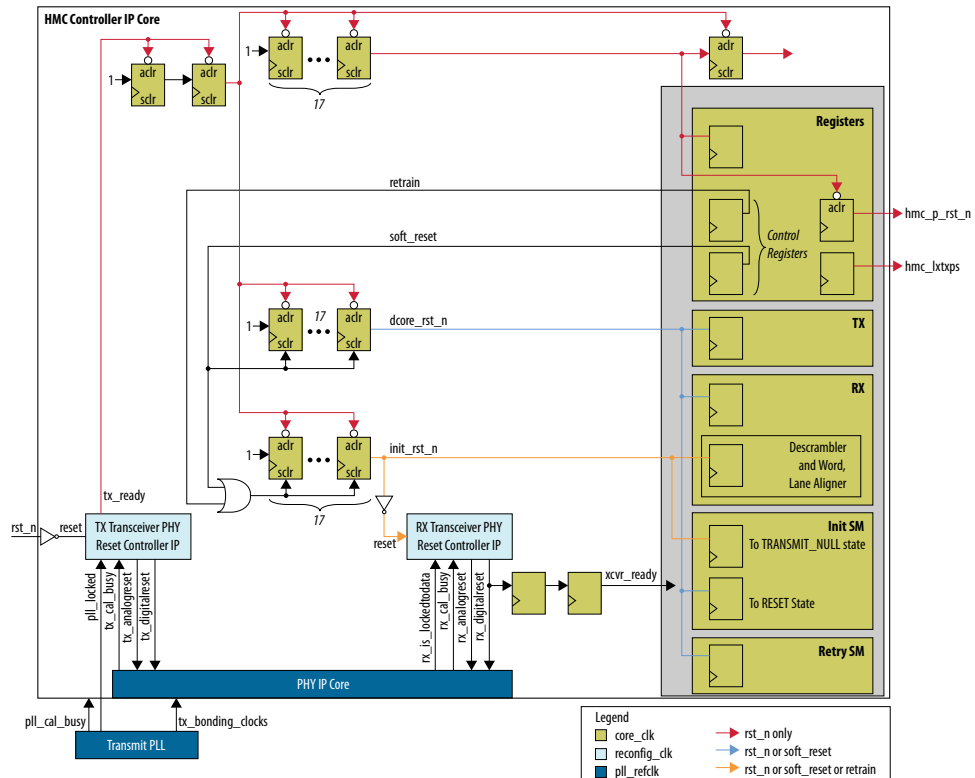
クロック信号とリセット信号、およびそれらの間の関係性についてリストします。

3.4 初期化とリセット

アクティブ Low `rst_n` 信号をアサートすると、HMC コントローラー IP コア、HMC デバイス、およびそれらに接続する HMC リンクの初期化がトリガーされます。正しいシーケンスを確保するために、HMC コントローラー IP コア、I²C マスターモジュール、および HMC デバイスを正しく接続する必要があります。IP コアは、リンクの再初期化、ソフトリセット、致命的なエラー回復、およびパワー・マネージメントのスリープおよびダウンモードをサポートするレジスターフィールドを提供します。

図 -12: Stratix 10 の HMC コントローラー IP コアの初期化およびリセットのオプション

HMC コントローラー IP コアは、HMC コントローラー IP コアの CONTROL レジスターの `rst_n` 入力信号や `SoftReset` フィールドおよび `Retrain` フィールドなど、IP コアのさまざまな部分をリセットするためのいくつかのメカニズムを提供しています。



初期化

次の信号は HMC コントローラー IP コア、HMC リンク、および HMC デバイスの初期化を制御します。

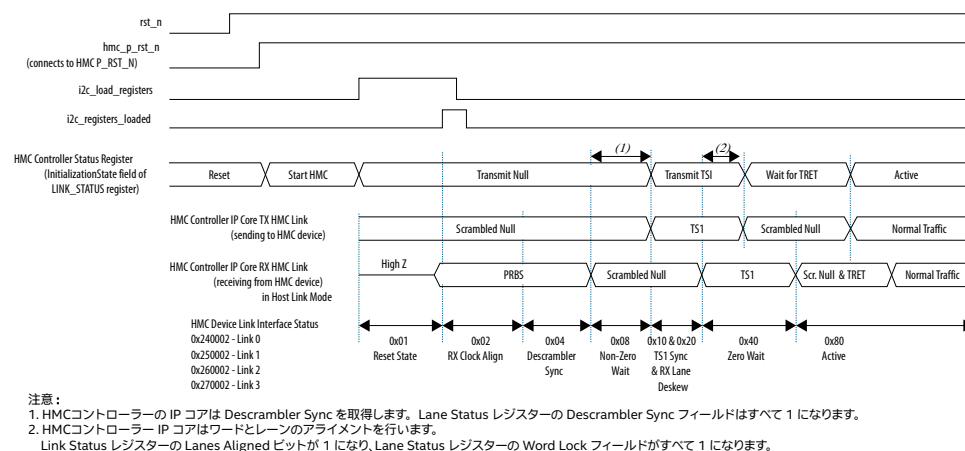
- `rst_n`: IP コアの初期化をトリガーするアクティブ Low の HMC コントローラー IP コア入力信号。
- `hmc_p_rst_n`: HMC コントローラー IP コアの出力信号。
 - 制御信号 : `rst_n` がアサートされ、それがデアサートされた直後に、IP コアはこの出力信号を制御します。ただし、`rst_n` がデアサートされた際に IP コアがこの信号を発生した後に、この信号は CONTROL レジスターのビット [17] 内の `P_RST_N` フィールドに駆動されます。ソフトウェアはこのビットを変更して HMC デバイスのリセットを強制することができます。
 - 接続信号 : この信号をアクティブ Low の HMC デバイスの `P_RST_N` 入力信号に接続する必要があります。`P_RST_N` は HMC デバイスの初期化をトリガーします。
- `i2c_load_registers`: HMC コントローラー IP コア の出力信号。この信号を I²C マスターモジュールの入力信号に接続し、HMC デバイスのコンフィグレーション・レジスターをロードするように I²C マスターモジュールに通知する必要があります。
- `I2c_registers_loaded`: I²C マスターモジュールが HMC デバイスの初期化でその部分を完了したことを示す HMC コントローラー IP コアの出力信号。HMC コンフィグレーション・レジスターのロード・シーケンスが正常に終了したことを示すように、この信号を I²C マスターモジュールの出力信号に接続する必要があります。

IP コアは、オフセット 0x10 にある `LINK_STATUS` レジスターの `InitializationState` フィールドにリンク初期化ステータスを報告します。HMC デバイスは、独自のリンク・インターフェイス・ステータスレジスターに独自のリンク初期化ステータスを報告します。

次の HMC コントローラー IP コア要件を考慮して、HMC リンクを初期化します。

- HMC コントローラー IP コアは Response Open Loop モードで動作し、このモードで HMC デバイスが IP コアと正しく通信するようにコンフィグレーションする必要があります。
- IP コアは `TGA` フィールドをサポートしていないため、ライト応答パケットですべてのノンポストド Write 要求に応答するように HMC デバイスをコンフィグレーションする必要があります。

図 -13: HMC リンク初期化シーケンス





致命的なエラー回復

HMC デバイスが致命的なエラーを宣言した場合は、HMC デバイスのウォームリセットと HMC コントローラー IP コアのリセットを実行する必要があります。HMC デバイスのウォームリセットを実行するには、HMC Global Configuration レジスターで適切なフィールドを設定します。

HMC コントローラー IP コアのソフトリセットまたはハードリセットを実行することができます。HMC コントローラー IP コアのソフトリセットを実行するには、HMC コントローラー IP コアの CONTROL レジスターのビット [2] の SoftReset フィールドを設定します。ハードリセットを実行するには、rst_n 入力信号をアサートします。

HMC コントローラー IP コアが致命的なエラーを宣言した場合は、CONTROL レジスターのビット [1] の ClearFatalError フィールドを設定します。場合により、このアクションは IP コアの通常の動作の再開を強制するには十分です。このアクションで不十分な場合は、IP コアを完全に再初期化する必要があります。

パワー・マネジメント

リンク上で非アクティブ中に節電するには、HMC リンクをスリープモードにします。スリープモードで一定時間が経過すると、HMC デバイスは自動的にダウンモードに移行します。

リンクをスリープ状態にするには次の手順を実行します。

1. データパス要求インターフェイスで要求の送信を停止し、IP コアの HMC リンク上の要求送信を強制的に停止するようにします。
2. すべての応答が到着し、トークンが返るのを待機します。
3. CONTROL レジスターのビット [16] の TXPS フィールドに 0 の値を書き込みます。
4. IP コアが LINK_STATUS レジスターのビット [16] の RXPS フィールドを 0 の値に設定するのを待機します。

スリープモードまたはダウンモードからリンクを起動するには、次の手順を実行します。

1. CONTROL レジスターのビット [0] の Retrain フィールドに 1 の値を書き込みます。
2. CONTROL レジスターのビット [16] の TXPS フィールドに 1 の値を書き込みます。
3. IP コアが LINK_STATUS レジスターのビット [16] の RXPS フィールドを 1 の値に設定するのを待機します。

リンクの再初期化

致命的なエラーの回復、スリープモードからのリンク起動、または高位ビット・エラーレートを示すリンクの再トレーニングするには、HMC リンクを再初期化する必要があります。

HMC デバイスがスリープモードを終了した後、ワードとレーンのアライメントを実行するためにデスクランブラーを再同期させてリンクを再初期化する必要があります。

リンクを再初期化するには、CONTROL レジスターのビット [0] の Retrain フィールドに 1 を書き込みます。

関連情報

- 24 ページの [外部 I2C マスターモデルの追加](#)
I²C マスターモジュールの要件および HMC コントローラー IP コアへのモジュールの接続について説明します。

- 46 ページの [I2C マスターへのインターフェイス上の信号](#)
初期化中の I²C マスターモジュールの詳細な動作について説明します。
- 44 ページの [HMC インターフェイス信号](#)
このインターフェイスは hmc_p_rst_n 信号を含んでいます。
- 48 ページの [クロック信号およびリセット信号](#)
- 53 ページの [LINK_STATUS レジスター](#)
- 52 ページの [CONTROL レジスター](#)

3.5 M20K ECC サポート

HMC コントローラー IP コア・バリエーションで **Enable M20K ECC support** を有効にすると、IP コアはデバイスの IP コアで構成されているすべての M20K ブロック内の ECC チェック用における内蔵デバイスサポートを活用します。この機能は、IP コアで構成された M20K メモリーブロックで、1 ビット・エラー訂正、隣接する 2 ビット・エラー訂正、隣接する 3 ビット・エラー訂正を実行します。

Stratix 10 の M20K メモリーブロックの ECC 機能の詳細については、*Stratix 10 Embedded Memory User Guide* を参照してください。

HMC コントローラー IP コアは、オフセット 0x38 にある `RETRY_BUFFER_ECC_COUNT` レジスターとオフセット 0x3C にある `RESPONSE_QUEUE_ECC_COUNT` レジスターに ECC エラー統計を報告します。

この機能ではデータの信頼性が向上しますが、request-to-response レイテンシーとリソース使用率が増加します。この機能を有効にすると、最大動作周波数 (f_{MAX}) が低下し、タイミング終了がより難しくなる可能性があります。

関連情報

58 ページの [エラーおよびリトライ・スタティック・レジスター](#)

`RETRY_ECC_COUNT` および `RESPONSE_ECC_COUNT` レジスターについて説明します。

3.6 フロー制御

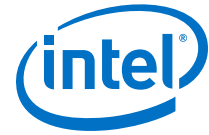
HMC 仕様には、トークンベースのフロー制御および Response Open Loop モードのホストと HMC メモリーデバイス間のトラフィックにおける 2 つの可能なフロー制御手法が説明されています。

Token-passing モードでは、ホストはトランザクション・レイヤーの初期化中に HMC デバイスにそのバッファリング容量に関する情報を送信します。Response Open Loop モードでは、ホストはバッファリング容量に関する情報を HMC デバイスに送信しません。代わりに、ホストはいつでも応答を受信する余地がある場合にのみ要求パケットを送信します。

HMC コントローラー IP コアは、Response Open Loop モードで動作します。IP コアは、HMC デバイスからのすべての応答パケットを受け入れる能力があるように設計されています。

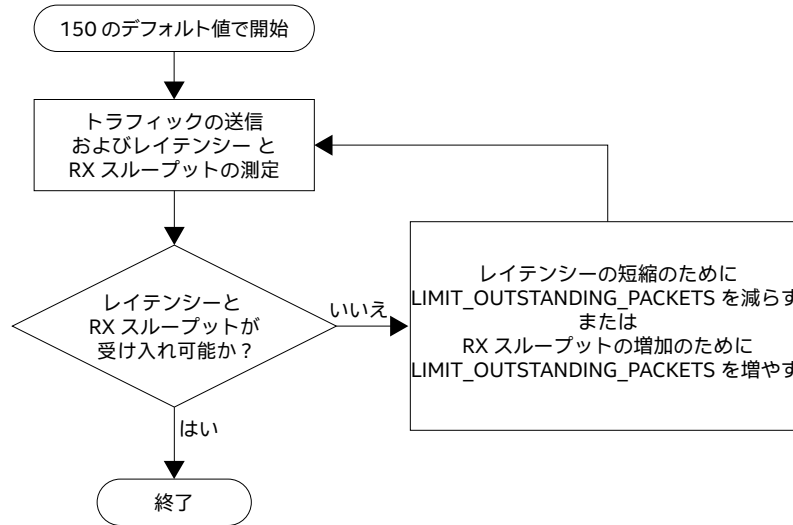
HMC コントローラー IP コアより速く届くユーザー要求が HMC リンク上に送信する際、HMC コントローラー IP コアは `dp_req_ready` 信号をディアサートすることでアプリケーションを背圧します。

さらに、IP コアは Limit Outstanding FLIT 機能をサポートしています。この機能を有効にすると、IP コアは予期されるリード応答の未解決の FLIT 数を制限するようになります。IP コアは、HMC から保留中の応答 FLIT 数をユーザー指定のしきい値以上に増加させると、HMC に要求の送信を遅延させます。



Limit Outstanding FLITs 機能は、応答パケットのペイロードが大きい場合に発生する輻輳を回避することで、システム内のパケットのワーストケースのラウンドトリップ・レイテンシーを制限します。例えば、RD128 要求のシーケンスは応答パスに輻輳を発生させます。リードインテンシブでレイテンシー重視のアプリケーションを構築する設計者は、Limit Outstanding FLITs 機能でレイテンシーとスループットを向上させることができます。

図 -14: レイテンシーとスループット改善のフローチャート



関連情報

54 ページの [LIMIT_OUTSTANDING_PACKETS レジスター](#)

3.7 エラー検出および管理

HMC 仕様では、エラー検出と回復プロセスを定義しています。HMC コントローラー IP コアはこれらの要件に準拠し、エラー管理をサポートするために次の追加機能を実装しています。

- 絶えず連続して到着するエラー応答を削除せずにソフトウェア処理をサポートするエラー応答キュー
- ささまざまなエラーカテゴリーのパケットの数を数える統計レジスター

表 9. HMC 応答パケット・フィールド・チェック

HMC コントローラーはエラーを示すこれらの HMC パケットフィールドをチェックし、HMC デバイスにパケットの再送を強制する Error Abort モードを入力してエラーを処理します。このモードでは、IP コアは部分的に送信されたパケット送信を完了し、各 HMC 仕様に従って IRTRY パケットを送信します。また、IP コアは指示されたビットを INTERRUPT_STATUS レジスターに設定し、LOCAL_ERROR_COUNT レジスターの Local Count フィールドをインクリメントします。

受信されるパケットフィールド	エラー表示	INTERRUPT_STATUS レジスタービット
LNG, DLN	両フィールドで異なる値、または無効な値	LNG/DLN Error
CRC	不正確な CRC	CRC Error
SEQ	予期しない値	SEQ Error

HMC コントローラー IP コアは ERRSTAT フィールド値もチェックし、次の規則に従って応答を処理します。

- ERRSTAT がゼロの値の場合、このフィールドにはエラーまたは条件がないことを示します。IP コアは応答パケットを通常通り処理します。
- ERRSTAT がリード応答、ライト応答または MODE 応答パケットでゼロ以外の値の場合、IP コアは通常通り応答を処理しますが、応答をアプリケーションに渡す際に RX データパス・インターフェイスに dp_rsp_error 信号をアサートします。
- ERRSTAT が応答パケットでゼロ以外の場合、IP コアは エラー応答パケットを RX データパス・インターフェイスに転送しません。代わりに、IP コアはパケットの ERRSTAT およびキューブ ID 値を内部エラー応答 FIFO に転送します。内部エラー応答 FIFO の最初のエレメントは、常に ERROR_RESPONSE レジスターに読み出し可能です。これらのパケットはソフトウェアで処理できます。

HMC コントローラー IP コアは、再トレーニングのたびに 32 個の IRTRY パケットを送信します。

注意: IP コアは HMC デバイスから少なくとも 20 個の IRTRY パケットの受信を予期します。

関連情報

- 55 ページの [割り込み関連のレジスター](#)
INTERRUPT_STATUS 割り込みビット CRC Error、SEQ Error、および LNG/DLN Error について説明します。
- 58 ページの [エラーおよびリトライ・スタティック・レジスター](#)
LOCAL_ERROR_COUNT レジスターの Local Count フィールドについて説明します。
- 54 ページの [ERROR_RESPONSE レジスター](#)
ERROR_RESPONSE レジスターおよびエラー応答キューについて説明します。

3.8 テスト機能

HMC コントローラー IP コアは複数のテスト機能をサポートしています。

- HMC コントローラー IP コアの CONTROL レジスターにフィールドを書き込むことにより、次のテスト機能を制御できます。
 - HMC コントローラー IP コアに入カストリームのエラーの検出と HMC デバイスへの StartRetry 要求の送信を強制します。
 - IP コアが送信する次の要求パケットの CRC に 1 ビット・エラーを注入します。
 - Retry State Machine に致命的なエラー状態の終了を強制します。
 - HMC デバイスと IP コアにリセットを強制します。
- ネイティブ PHY IP コアが提供するテスト機能を使用できます。ハード PCS レジスターにフィールドを書き込んでこれらの機能を制御します。これらのレジスターへの書き込みアクセスは、トランシーバー・リコンフィグレーション・インターフェイスを介して行います。**Enable ADME and Optional Reconfiguration Logic** を有効にすると、システムコンソールからアクセス可能な JTAG マスターを介してこれらのレジスターに書き込みをアクセスできます。

関連情報

- 49 ページの [トランシーバー・リコンフィグレーション信号](#)
- 52 ページの [CONTROL レジスター](#)



4 HMC コントローラー IP コア信号

HMC コントローラー IP コアは、複数のインターフェイスを介して他のデザイン・コンポーネントと通信します。IP コアには、次のトップレベルの信号があります。

- 37 ページの [アプリケーション・インターフェイス信号](#)
- 44 ページの [HMC インターフェイス信号](#)
- 46 ページの [I2C マスターへのインターフェイス上の信号](#)
- 47 ページの [コントロール・レジスター・インターフェイス信号およびステータス・レジスター・インターフェイス信号](#)
- 48 ページの [ステータス信号およびデバッグ信号](#)
- 48 ページの [クロック信号およびリセット信号](#)
- 49 ページの [トランシーバー・リコンフィグレーション信号](#)
- 50 ページの [外部 PLL へのインターフェイス上の信号](#)

4.1 アプリケーション・インターフェイス信号

アプリケーション・インターフェイスは、メモリーの Read および Write 要求を指定して Read およびライト応答を受信するための単一データバス・インターフェイスを提供することで、外部 HMC への容易なアクセスをサポートします。このインターフェイスはデータバス・インターフェイスとも呼ばれます。

HMC コントローラーは 1 つ、2 つ、3 つ、または 4 つのデータバス・インターフェイスをサポートしています。HMC コントローラーのパラメーター・エディターのデータバス・インターフェイスの数を設定します。

関連情報

- 28 ページの [アプリケーション・インターフェイス](#)

4.1.1 アプリケーション要求インターフェイス

データバス要求インターフェイスまたはアプリケーション要求インターフェイスは、512 ビット・データバス、および HMC 要求パケットフィールド値を HMC コントローラー IP コアに提供するためのアプリケーション用の専用信号を提供します。インターフェイスは 128 バイトまでのペイロードサイズの Write 要求をサポートします。最大ペイロードサイズは、インターフェイスを 2 以下の core_clk クロックサイクルのデータバーストに制限します。バスサイズの倍数ではないペイロードサイズを有する Write 要求およびリード応答は、最後のクロックサイクルでデータバスの下位ビットにペイロードの終わりをキャリーします。

IP コアは、1 つ、2 つ、3 つ、または 4 つのデータバス要求インターフェイスを有することができます。インターフェイスはナンバーリングされており、**Response re-ordering** パラメーターをオフにすると、それぞれは設定されたタグ範囲に制限されます。この制限に反すると結果は未定義になります。

アプリケーションは、TX データパス・インターフェイスで送信するすべての要求に対して、次の配線および制御情報を提供する必要があります。

- 整形形式 HMC デスティネーション・アドレス
- 特有の 9 ビットの in-flight タグ (**Response re-ordering** をオフにした場合)。この要件はポストッド・トランザクション要求には適用されません。複数のポートの場合、各ポートには排他的に使用するための専用のタグ範囲が割り当てられます。アプリケーションはそのタグが有する前のトランザクションが完全に処理された後にのみ、タグを再使用できます。
- 正確な 3 ビット・キューブ ID

加えて、アプリケーションは応答が到着するとすぐに要求に対する応答が受信できる場合にのみ、要求を送信するようにする必要があります。

図 -15: シングルポートを有する HMC コントローラー IP コアへのアプリケーション

クライアントはソースとして動作し、HMC コントローラーは送信方向のシンクとして動作します。

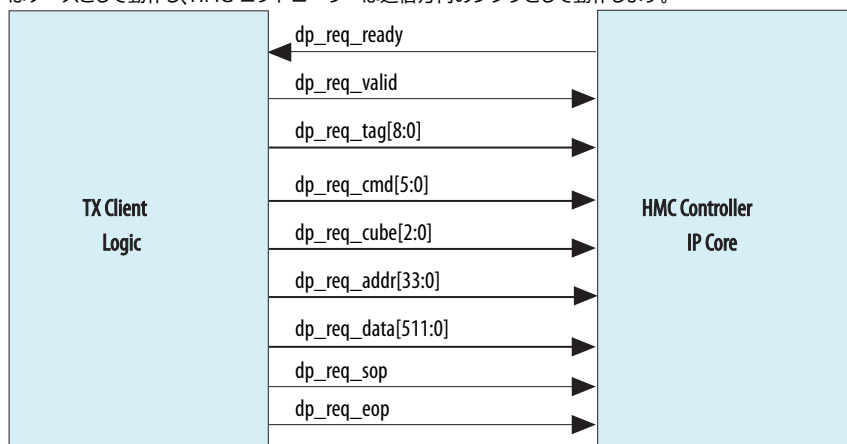


図 -16: 複数のポートを有する HMC コントローラー IP コアへのアプリケーション

クライアントは送信元として動作し、HMC コントローラーは送信方向のシンクとして動作します。N は **Ports** パラメーターで指定するポートの数です。

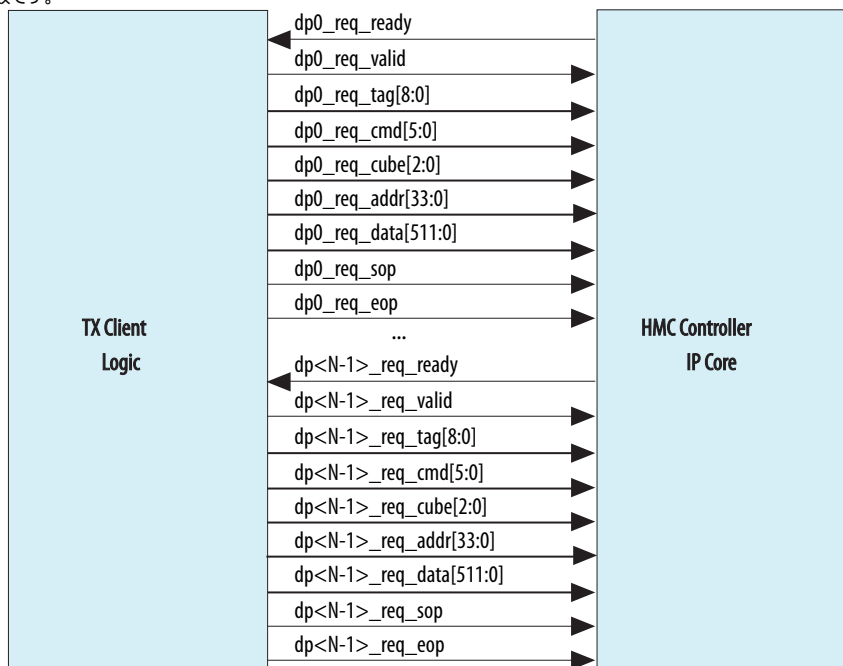


表 10. 各データパス要求インターフェースの信号

すべてのインターフェイス信号は `core_clk` クロックと同期しています。IP コアに複数のポートがある場合、ポートナンバーは図のように信号名の一部です。IP コアに 1 つのポートがある場合、IP コアの以前のリリースとの下位互換性のために、ポートナンバーは信号名を含みません。IP コアに 2 つのポートがある場合、`n=0` が 1 つと `n=1` が 1 つの 2 セットの信号があります。IP コアに 4 つのポートがある場合、`n=0, 1, 2, および 3` の 4 セットの信号があります。

信号名	入力 / 出力	説明
<code>dp<n>_req_ready</code>	出力	HMC コントローラー IP コアをこの信号にアサートすると、IP コアはデータの受信を準備します。 <code>dp<n>_req_ready</code> と <code>dp<n>_req_valid</code> の両方が同じサイクルでアサートされると、HMC コントローラー IP コアはデータを受け入れます。 IP コアはこの信号をデアサートし、現在の着信要求も処理できないときにアプリケーションに背圧をかけます。 IP コアはマルチサイクル書き込みデータ転送のサイクル間にこの信号をデアサートしません。
<code>dp<n>_req_valid</code>	入力	すべての入力信号に有効な値があり、トランザクションが有効であることを表示します。HMC コントローラー IP コアは、 <code>dp<n>_req_ready</code> と <code>dp<n>_req_valid</code> の両方がアサートされると、 <code>core_clk</code> の立ち上がりエッジでデータを受け入れます。 アプリケーションは、マルチサイクル書き込みデータの転送中にアサートされたこの信号を保持する必要があります。 アプリケーションは、アサートされた <code>dp<n>_req_valid</code> 信号を保持することで連続して要求を送信できます。 注 応答を必要とする要求の場合、アプリケーションは新しい要求に対して <code>dp<n>_req_tag</code> で駆動する値を更新する必要があります。アプリケーションが入力信号で古い値を続けて駆動し続ける場合、その新しい要求は前の要求と同じタグを有します。

continued...



信号名	入力 / 出力	説明
dp<n>_req_tag[8:0]	入力	<p>この要求に関連付いたユーザー生成のタグです。対応する応答は同じタグで返されます。</p> <p>Response re-ordering を有効にすると、この信号は使用できません。その場合、IP コアはタグを内部的に管理します。</p> <p>この信号の値はポストド・トランザクション要求に対する Don't Care です。これらの要求には対応する応答がないため、意味のあるタグの値は不要です。</p> <p>特定の時点で使用されているすべてのタグがノンポストド・トランザクションにより一意であることを保証する必要があります。応答が返った後、タグを再利用できます。</p> <p>IP コアに複数のポートがある場合、各ポートは特定のタグの範囲に制限されます。</p> <ul style="list-style-type: none"> ポート 2 つ：ポート 0 のタグ範囲は 0 ~ 255 で、ポート 1 のタグ範囲は 256 ~ 511 ポート 3 つ：ポート 0 のタグ範囲は 0 ~ 175、ポート 1 のタグ範囲は 176 ~ 351、ポート 2 のタグ範囲は 352 ~ 511 ポート 4 つ：ポート 0 のタグ範囲は 0 ~ 127、ポート 1 のタグ範囲は 128 ~ 255、ポート 2 のタグ範囲は 256 ~ 383、ポート 3 のタグ範囲は 384 ~ 511 <p>この制限に反すると、結果は未定義となります。</p> <p>アプリケーションは、マルチサイクル書き込みデータの転送中、この信号の値を保持する必要があります。</p>
dp<n>_req_cmd[5:0]	入力	<p>この要求に関連付けられたパケットコマンドを示します。コマンドのエンコードについては、HMC 仕様 v1.1 の表 17 を参照してください。</p> <p>アプリケーションは、マルチサイクル書き込みデータの転送中、この信号の値を保持する必要があります。</p>
dp<n>_req_cube[2:0]	入力	<p>要求が送信されるキューブの CUB ID。</p> <p>アプリケーションは、マルチサイクル書き込みデータの転送中、この信号の値を保持する必要があります。</p>
dp<n>_req_addr[33:0]	入力	<p>外部 HMC デバイスのターゲットアドレスです。現在の HMC デバイスは、HMC コントローラー IP コアが生成する BIT WRITE 要求を除いたすべての要求でアドレスの 4 つの最下位ビット (4'b0000 の値を有すると仮定) を無視します。</p> <p>アプリケーションは、マルチサイクル書き込みデータの転送中、この信号の値を維持する必要があります。</p>
dp<n>_req_data[511:0]	入力	<p>書き込みデータ。</p> <p>アプリケーションは、最初のサイクルで書き込みペイロードの最下位バイトを転送する必要があります。</p> <p>ペイロードのサイズがデータバス幅の整数倍でない場合、アプリケーションは最後のデータ転送サイクルで残りの書き込みペイロードを dp<n>_req_data または dp_req_data の最下位バイトに転送する必要があります。例えば、アプリケーションは次を転送します。</p> <ul style="list-style-type: none"> dp_req_data[127:0] に 16 バイトを転送 dp_req_data[255:0] に 32 バイトのペイロードを転送 2 番目のデータ転送クロックサイクルで、dp_req_data[383:0] に 112 バイトのペイロードの最後の (最も上位の) 48 バイトを転送 <p>Read 要求中、このデータバスの値は重要ではありません。</p>
dp<n>_req_sop	入力	<p>パケットの開始。アプリケーションは、すべてのトランザクションの最初のサイクルでこの信号をアサートする必要があります。</p>
dp<n>_req_eop	入力	<p>パケットの終了。アプリケーションは、すべてのトランザクションの最終サイクルでこの信号をアサートする必要があります。</p>

4.1.2 アプリケーション応答インターフェイス

データパス応答インターフェイスまたはアプリケーション応答インターフェイスは、512 ビット・データバス、および HMC 応答情報をアプリケーションに提供するための IP コア用の専用信号を提供します。インターフェイスは 128 バイトまでのペイロードサイズのリード応答をサポートします。最大ペイロードサイズは、インターフェイスを 2 以下の `core_clk` クロックサイクルのデータバーストに制限します。バスサイズの倍数ではないペイロードサイズを有するリード応答は、最後のクロックサイクルでデータバスの下位ビットにペイロードの終わりをキャリーします。

Response re-ordering をオフにすると、HMC コントローラーはデータパス応答インターフェイスで送信するすべての応答とともに、元の要求から 9 ビット・タグを返します。アプリケーションは、各応答を対応する要求と照合するためにタグを使用します。

IP コアのデータパス応答インターフェイスを背圧することはできません。受信するすべての応答をアプリケーションが確実に処理できるようにするには、アプリケーションは応答を処理またはバッファードするためのリソースを有する要求のみを送信する必要があります。

図 -17: **RX アプリケーションへのシングルポートを有する HMC コントローラー IP コア**

HMC コントローラー IP コアはソースとして動作し、クライアントは受信方向のシンクとして動作します。

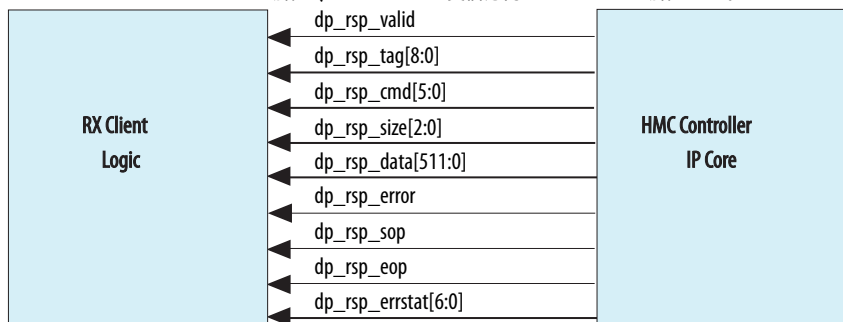


図 -18: RX アプリケーションへの複数のポートを有する HMC コントローラー IP コア

HMC コントローラー IP コアはソースとして動作し、クライアントは受信方向のシンクとして動作します。N は **Ports** パラメータで指定するポートの数です。

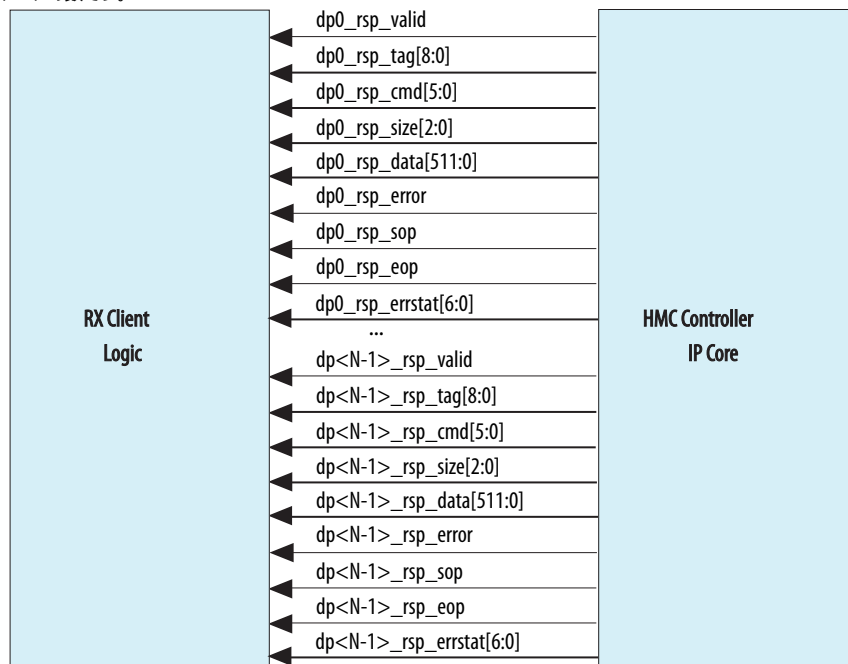


表 11. 各データバス応答インターフェイスの信号

すべてのインターフェイス信号は core_clk クロックによりクロックされます。IP コアが複数のポートの場合、ポートナンバーは図のように信号名の一部です。IP コアに 1 つのポートがある場合、IP コアの以前のリリースとの下位互換性のために、ポートナンバーは信号名を含みません。IP コアに 2 つのポートがある場合は、n=0 が 1 つと n=1 が 1 つの 2 セットの信号があります。IP コアに 3 つのポートがある場合、n=0、1、および 2 の 3 セット信号があります。IP コアに 4 つのポートがある場合、n = 0、1、2、および 3 の 4 セットの信号があります。

信号名	入力 / 出力	説明
dp<n>_rsp_valid	出力	dp<n>_rsp_tag, dp<n>_rsp_cmd, dp<n>_rsp_error, dp<n>_rsp_sop, dp<n>_rsp_eop, および dp<n>_rsp_errstat 信号のすべてが有効であることを示し、パイロードのリード応答で dp<n>_rsp_data と dp<n>_rsp_size が有効です。 アプリケーションはすべての有効なトランザクションを受け入れる必要があります。HMC コントローラー IP コアのデータバス応答インターフェイスを背圧することはできません。 IP コアは、マルチサイクル読み出しデータの転送の間にこの信号をアサートしません。
dp<n>_rsp_tag[8:0]	出力	応答への元の要求に関連付けられたタグ。 この応答を処理した後、タグを再使用できます。 IP コアはマルチサイクル読み出しデータの転送中、この信号の値を保持します。 応答の Response re-ordering を有効にすると、この信号は使用できません。この場合、IP コアは内部的にタグを管理します。
dp<n>_rsp_cmd[5:0]	出力	この応答に関連するパケットコマンドを示します。コマンドのエンコーディングについては、HMC 仕様 v1.1 の表 25 を参照してください。この信号は、エラーのない応答コードのみを保持し、つまり IP コアはエラー応答をレジスターに配線します。 IP コアは、マルチサイクル読み出しデータ転送の間、この信号の値を保持します。

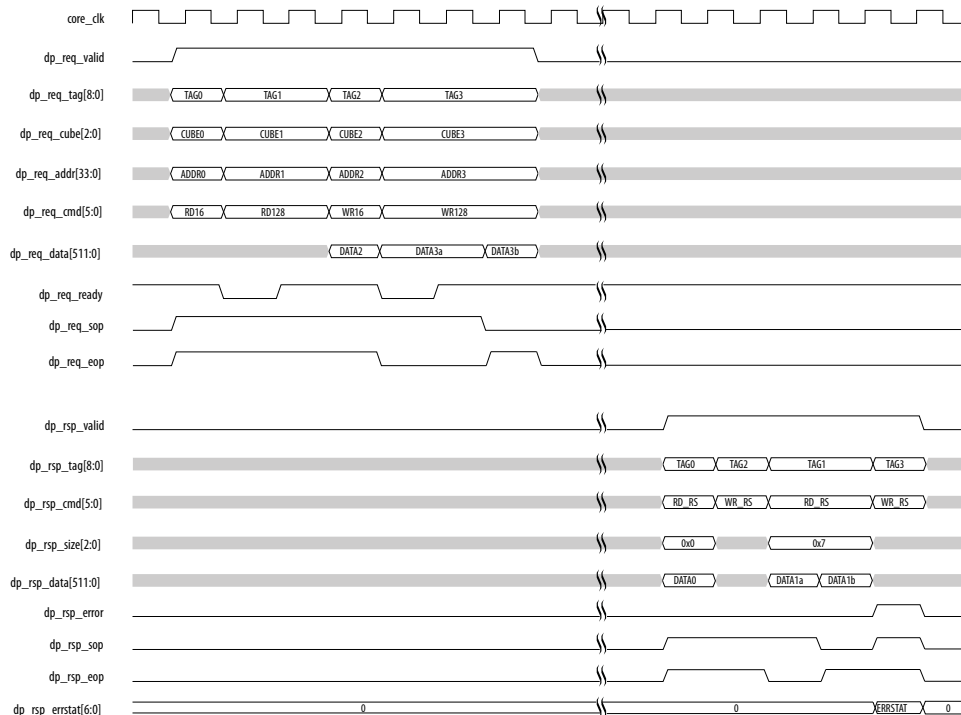
continued...



信号名	入力 / 出力	説明
dp<n>_rsp_size[2:0]	出力	<p>この応答に関連付けられたペイロードのサイズを示します。現在の応答が リード 応答の場合は、ペイロードのサイズを dp<n>_rsp_data に表示します。関連するペイロードの応答中、IP コアはこの信号を次の有効な値のいずれかに設定します。</p> <ul style="list-style-type: none"> 3'b000 は 16 バイトのペイロードを表示 3'b001 は 32 バイトのペイロードを表示 3'b010 は 48 バイトのペイロードを表示 3'b011 は 64 バイトのペイロードを表示 3'b100 は 80 バイトのペイロードを表示 3'b101 は 96 バイトのペイロードを表示 3'b110 は 112 バイトのペイロードを表示 3'b111 は 128 バイトのペイロードを表示 <p>ペイロードが関連付けられていない応答の間、この信号の値は未定義です。関連するペイロードがない応答は、dp<n>_rsp_cmd[0]の値が 1 の応答です。IP コアはマルチサイクル読み出しデータ転送中、この信号の値を維持します。</p>
dp<n>_rsp_data[511:0]	出力	<p>リード応答データ。</p> <p>ペイロードが関連付けられていない応答の間、この信号の値は未定義です。関連するペイロードがない応答は、dp<n>_rsp_cmd[0]の値が 1 の応答です。ペイロードのサイズがデータバス幅の整数倍でない場合、最後のデータ転送サイクルで IP コアは残りのリードペイロードを dp<n>_rsp_data または dp_rsp_data の最下位バイトに転送します。例えば、IP コアは次を転送しません。</p> <ul style="list-style-type: none"> dp_rsp_data[127:0]に 16 バイトのペイロードを転送 dp_rsp_data[255:0]に 32 バイトのペイロードを転送 2 番目のデータ転送クロックサイクルで dp_rsp_data[383:0] のペイロードで 112 バイトの最終 (最も上位の) 48 バイトを転送します。
dp<n>_rsp_error	出力	<p>対応する要求がエラーで完了し、自動的に再試行されないことを表示します。HMC コントローラー IP コアは、外部 HMC デバイスからゼロ以外の の ERRSTAT または DINV フィールドを使用して Read または ライト応答パケットを受信すると、この信号をアサートします。IP コアはマルチサイクル読み出しデータ転送中、この信号の値を保持します。</p>
dp<n>_rsp_sop	出力	<p>パケットの開始。IP コアはすべての応答トランザクションの最初のサイクルでこの信号をアサートします。</p>
dp<n>_rsp_eop	出力	<p>パケットの終わり。IP コアはすべての応答トランザクションの最後のサイクルでこの信号をアサートします。</p>
dp<n>_rsp_errstat[6:0]	出力	<p>エラーステータス。IP コアは外部 HMC デバイスの ERRSTAT フィールドからこの値を直接渡します。</p>

4.1.3 HMC コントローラー IP コアのデータパス例

図 -19: HMC コントローラー IP コアのアプリケーション・インターフェイス例



この例では、ユーザーロジックは、シングルデータ・インターフェイス・ポートを有する HMC コントローラー IP コア 4 つの連続した要求パケットを送信します。IP コアは対応する応答パケットを送信します。最初の 3 つの要求はエラーなしで完了します。4 番目の要求はエラー表示で完了します。

HMC コントローラー IP コアが dp_req_ready 信号をディアサートすると、ユーザーロジックは IP コアが dp_req_ready を再アサートした後にフル・クロックサイクルまで現在の値を保持します。ユーザーロジックは、dp_req_ready がアサートされている間に値を送信する必要があり、IP コアが正しくそれらをキャプチャーするようにします。

IP コアは、WR128 要求への応答を送信しながら、dp_rsp_error 信号をアサートします。この信号は WR128 要求が正常に終了しなかったことを表示します。IP コアはこの情報を HMC デバイスから渡します。したがって、この信号は HMC デバイスが要求を処理しようとしている間にエラーを検出したことを表示します。

4.2 HMC インターフェイス信号

HMC コントローラー IP コアの HMC インターフェイスは、外部 HMC デバイスのリンク・インターフェイスとメインのリセット信号に接続します。

表 12. HMC インターフェイスの信号

信号名	入力 / 出力	説明
hmc_lxrx[15:0]	入力	受信レーン。HMC 仕様の LxRXP と LxRXN の差動ペアを実装します。
<i>continued...</i>		



信号名	入力 / 出力	説明
		このデータバスを HMC デバイスの LxTXP バスに接続する必要があります。Quartus Prime Fitter は正しいピンを負の信号に自動的に割り当てます。
hmc_lxtx[15:0]	出力	送信レーン。HMC 仕様の LxTXP と LxTXN の差動ペアを実装します。このデータバスを HMC デバイスの LxRXP バスに接続する必要があります。Quartus Prime Fitter は正しいピンを負の信号に自動的に割り当てます。
hmc_lrxrps	入力	リンク電力低減入力。HMC 仕様の LxRXPS 信号を実装します。HMC コントローラー IP コアは LINK_STATUS レジスタの RXPS フィールドの値をこの信号の値に設定します。この入力信号を HMC デバイスの LxTXPS 出力信号に接続する必要があります。
hmc_lxtxps	出力	リンク電力低減出力。HMC 仕様の LxTXPS 信号を実装します。IP コアは CONTROL レジスタの TXPS ビットの値でこの信号を駆動します。出力信号を HMC デバイスの LxRXPS 入力信号に接続します。
hmc_ferr_n	入力	HMC デバイスからのアクティブ Low の致命的なエラー表示。HMC コントローラー IP コアは設定します。HMC コントローラー IP コアは、INTERRUPT_STATUS レジスタの FERR_N フィールドの値をこの信号の値に設定します。この信号を HMC デバイスの FERR_N 信号に接続する必要があります。
hmc_p_rst_n	出力	HMC デバイスへのメインのリセット信号。IP コアは、この信号を CONTROL レジスタの P_RST_N ビットの値で駆動します。この信号を HMC デバイスの P_RST_N 信号に接続する必要があります。

関連情報

- 31 ページの [初期化とリセット](#)
初期化およびパワー・マネージメントで、HMC インターフェイス信号のリセットおよびパワー・マネージメントの使用を説明しています。
- 52 ページの [CONTROL レジスタ](#)
- 53 ページの [LINK_STATUS レジスタ](#)
- 55 ページの [割り込み関連のレジスタ](#)
INTERRUPT_STATUS レジスタについて説明します。
- [Hybrid Memory Cube Specification 1.1](#)
HMC 仕様は、ハイブリッド・メモリー・キューブ・コンソーシアム (HMCC) のウェブページからダウンロードできます。

4.3 I²C マスターへのインターフェイス上の信号

デザインには、リンク初期化のために HMC デバイスの I²C インターフェイスを駆動する I²C マスターモジュールが含まれている必要があります。このインターフェイスは I²C モジュールに接続します。

I²C モジュールと IP コアは、下の表の 2 つのインターフェイス信号を使用して、次の 4 方向のハンドシェイクを実装する必要があります。

1. IP コアをリセットすると、i2c_load_registers 信号がディアサートされます。I²C マスターモジュールをリセットすると、i2c_registers_loaded 信号がディアサートされます。
2. IP コアと HMC の準備が完了すると、IP コアは i2c_load_registers をアサートします。シミュレーションでは、IP コアは HMC シミュレーション・モデルが瞬時に準備されることが前提であり、ハードウェアでは IP コアは必要とする t_{INIT} の 20 ms の間、待機します。
3. I²C マスターモジュールは i2c_load_registers のアサーションを検出すると、HMC デバイスのレジスターに書き込みを行い、リンクの初期化 (Init Continue と結合する) を設定し、i2c_registers_loaded 信号をアサートします。
4. HMC コントローラー IP コアは i2c_load_registers をディアサートします。
5. I²C マスターモジュールは i2c_registers_loaded をディアサートします。

表 13. 外部 I²C マスターモジュールへのインターフェイス上の信号

IP コアの i2c_load_registers 信号の動作は、4 方向のハンドシェイク・プロトコルに準拠しています。HMC コントローラー IP コア機能を正しく設定するには、この 4 方向のハンドシェイク・プロトコルに準拠した i2c_registers_loaded 信号の動作を実装するために、デザインに I²C マスターモジュールを設計する必要があります。

信号名	入力 / 出力	説明
i2c_load_registers	出力	リンク初期化シーケンスの一部として、HMC コントローラー IP コアが外部 I ² C マスターモジュールが HMC デバイスのコンフィグレーション・レジスターをロードする準備が完了していることを表示します。 この信号は、HMC デバイスのコンフィグレーション・レジスターをロードする要求を受け入れる I ² C マスターモジュールの入力ポートに接続する必要があります。
i2c_registers_loaded	入力	外部 HMC デバイスのレジスターがコンフィグレーションされていることを表示します。 この信号は、コンフィグレーション・レジスターのロードシーケンスが正常に完了したことを表示する I ² C マスターの出力ポートに接続する必要があります。

複数の HMC コントローラー IP コアが同じ HMC デバイスの異なるリンクに接続されている場合、外部 I²C マスターは HMC デバイスのコンフィグレーション・レジスターに書き込む前に、すべての HMC コントローラー IP コアが i2c_load_registers 信号をアサートするまで待機する必要があります。外部 I²C マスターが HMC コンフィグレーション・レジスターのすべての書き込みを完了した後、すべての HMC コントローラー IP コアの i2c_registers_loaded 信号を同時にアサートする必要があります。

関連情報

60 ページの HMC コントローラー IP コア Stratix 10 のデザイン例

HMC コントローラーのデザイン例は、HMC コントローラー IP コアで 4 方向のハンドシェイク・プロトコルを正しく実装し、Micron HMC 15G SR HMC デバイスを初期化するためのレジスタ書き込みの推奨するシーケンスを実装する I²C マスターモジュールを含んでいます。



4.4 コントロール・レジスター・インターフェイス信号およびステータス・レジスター・インターフェイス信号

コントロールおよびステータス・レジスター・インターフェイスは、HMC コントローラー IP コアの内部コントロールおよびステータスレジスターへのアクセスを提供する Avalon-MM インターフェイスです。このインターフェイスはトランシーバー・コンフィグレーション・レジスターへのアクセスを提供しません。

Avalon-MM インターフェイスは標準のメモリーマップド・プロトコルを実装します。組み込み式プロセッサや JTAG Avalon マスターといった任意の Avalon マスターをこのバスに接続し、IP コアのコントロールおよびステータスレジスターへアクセスできます。

表 14. コントロールおよびステータス・レジスター・インターフェイス信号

core_clk は HMC コントローラー IP コアのコントロールおよびステータス・レジスター・インターフェイス上で信号をクロックします。このインターフェイスは 32 ビットのペイロード (1 つのフルレジスター値) のリード動作とライト動作のみをサポートします。

信号名	入力 / 出力	説明
csr_address[5:0]	入力	レジスターをリードおよびライトするバイトアドレス。すべての HMC コントローラーのコントロールおよびステータスレジスターは 32 ビット幅です。したがって、すべてのアドレスは 4 バイト境界です。
csr_read	入力	この信号をリード転送の要求にアサートする必要があります。
csr_write	入力	この信号をライト転送の要求にアサートする必要があります。
csr_writedata[31:0]	入力	ライトデータ
csr_readdata[31:0]	出力	リードデータ
csr_readdatavalid	出力	リードデータを使用する準備が完了したことを示します。
csr_irq	出力	割り込み要求。 この信号の値はこのインターフェイス上の他の信号の現在の値と関連付けられていません。IP コアは、INTERRUPT_STATUS レジスタービットがアサートされると (対応する INTERRUPT_ENABLE ビットが設定され、GLOBAL_INTERRUPT_ENABLE レジスターの GlobalEnable ビットが 1 の値を受け取ると) すぐにこの割り込み信号を非同期にアサートし、2 つの関連する有効なビットがリセットされるか、アプリケーションが現在アサートされているすべての INTERRUPT_STATUS レジスタービットに 1 の値を書き込むまで、この信号のアサートを保ちます。

関連情報

- 29 ページの [コントロール・レジスター・インターフェイスおよびステータス・レジスター・インターフェイス](#)
- 51 ページの [HMC コントローラー IP コアのレジスターマップ](#)
- 55 ページの [割り込み関連のレジスター](#)
INTERRUPT_STATUS、INTERRUPT_ENABLE、および
INTERRUPT_GLOBAL_ENABLE レジスターについて説明します。
- [Avalon Interface Specifications](#)
タイミング図を含む Avalon-MM プロトコルについての詳細は、*Avalon メモリーマップド・インターフェイス*の章を参照してください。

4.5 ステータス信号およびデバッグ信号

表 15. ステータスおよびデバッグ信号

HMC コントローラー IP コアのステータスおよびデバッグ・インターフェイスは、リンクの初期化と HMC システムのデバッグをサポートするためのいくつかの信号を提供しています。

クロック名	入力 / 出力	説明
link_init_complete	出力	IP コアはリンク初期化スタートマシンがアクティブ状態になると、この信号をアサートします。
debug_tx_data[511:0]	出力	このデータバスは、TX レーンスワッパーに入る前に、ストライプ化されたデータのスクランブルされていないコピーを示します。このバス上のデータはストライプ化されていますが、スクランブルされていません。
debug_rx_data[511:0]	出力	このデータバスは、RX レーンスワッパーおよびデスクランブラーによる処理の後、ストライプ化およびデスクランブルされた受信データを示します。各レーンでは、デスクランブラーが同期されていない場合はデスクランブラーの出力は強制的にゼロになります。このバスの値を解釈する前に、LANE_STATUS レジスタの DescramSync フィールドをリードすることにより、デスクランブラーの状態を確認します。

関連情報

29 ページの [ステータス・インターフェイスおよびデバッグ・インターフェイス](#)

4.6 クロック信号およびリセット信号

表 16. HMC コントローラー IP コアのクロック信号およびリセット信号

HMC コントローラー IP コアは、トランシーバーの外部に単一のクロックドメインを備えています。デザインでは、同じクロック・リファレンス・ソースからの外部 TX PLL リファレンス・クロック、RX CDR リファレンス・クロック、および HMC デバイスの REFCLKP および REFCLKN 入力リファレンス・クロック信号を派生させる必要があります。

クロック名	入力 / 出力	説明
rst_n	入力	HMC コントローラー IP コアのアクティブ Low マスターリセット信号。この信号は非同期です。アサートされると、IP コアがリセット要求をキャプチャーできるようにするために、信号は少なくとも 2 reconfig_clk クロックサイクルの間、アサート状態を保持する必要があります。
core_rst_n	出力	この信号がアサートされると、HMC コントローラー IP コアがリセットされていることを示します。IP コアは、core_clk が安定し、トランシーバーがデータを送信する準備ができた後にのみ、core_rst_n 信号をデアサートします。
rx_cdr_refclk0	入力	RX トランシーバー CDR PLL のリファレンス・クロック。このクロックは CDR reference clock パラメーターに指定する周波数で駆動する必要があります。rx_cdr_refclk0 は TX PLL のリファレンス・クロックではありません。TX PLL のリファレンス・クロックは、HMC コントローラー IP コアに接続する外部 TX PLL IP コアへの入力です。TX PLL のリファレンス・クロックは、HMC コントローラー IP コアを直接駆動しません。
tx_bonding_clocks[95:0]	入力	個別のトランシーバー・チャンネルのクロック。各トランシーバー・チャンネルへの入力クロックは 6 ビットです。この入力バスを外部トランシーバー TX PLL IP コアに接続する必要があります。レーンごとのデータレートの半分である出力周波数を指定するには、外部 TX PLL IP コアをパラメーター化する必要があります。HMC コントローラー IP コアの 10 Gbps レーンレートの場合、TX PLL IP コアの出力周波数は 5 GHz、12.5 Gbps レーンレートの場合 TX PLL IP コアの出力周波数は 6.25 GHz でなければなりません。また、15 Gbps レーンレートの場合、TX PLL IP コアの出力周波数は 7.5 GHz でなければなりません。

continued...



クロック名	入力 / 出力	説明	
core_clk	出力	HMC コントローラー IP コアのマスタークロック。トランシーバーは core_clk を生成します。core_clk の周波数はレーンレートを 32 で割ったものです。	
		レーンレート	core_clk 周波数
		10 Gbps	312.5 Mhz
		12.5 Gbps	390.625 MHz
		15 Gbps	468.75 MHz
core_clk はコントロールおよびステータスレジスター・インターフェイスの信号を含めて HMC コントローラー IP コア信号をクロックします。			
reconfig_clk	入力	トランシーバー・リコンフィギュレーション・インターフェイスのクロック。Stratix 10 デバイスをターゲットとする HMC コントローラー IP コアでは、内部リセット・コントローラーもクロックします。このクロックは、100 ~ 150 MHz の範囲の周波数で駆動する必要があります。	
reconfig_reset	入力	トランシーバー・リコンフィギュレーション・インターフェイスのリセット信号。この信号は非同期です。IP コアは reconfig_clk に同期してリセット要求をキャプチャーします。アサートされると、この信号は少なくとも 2 reconfig_clk サイクルの間、アサート状態を保持する必要があります。	

4.7 トランシーバー・リコンフィギュレーション信号

インテル は、トランシーバー・レジスターにアクセスするために、トランシーバー・リコンフィギュレーション・インターフェイスと呼ばれる専用の Avalon-MM インターフェイスを提供しています。IP コアの汎用コントロールおよびステータス・レジスター・インターフェイスを介してトランシーバー・レジスターにアクセスするのではなく、この専用インターフェイスを介してトランシーバー・レジスターにアクセスします。

注意: このインターフェイスは Quartus Prime Pro - Stratix 10 Edition Beta ソフトウェアに含まれていますが、このリリースでは機能しません。ただし、IP コアは内部的に reconfig_clk を使用します。IP コアが正しく機能するには、有効なクロックを reconfig_clk 入力ポートに接続する必要があります。

Avalon-MM インターフェイスは標準メモリーマップド・プロトコルを実装しています。Avalon マスターをこのバスに接続し、エンベデッド Native PHY IP コアのレジスターにアクセスすることができます。

表 17. HMC コントローラー IP コアのトランシーバー・リコンフィギュレーション・インターフェイス信号

Reconfig_clk は HMC コントローラー IP コアのトランシーバー・リコンフィギュレーション・インターフェイスで信号をクロックします。reconfig_reset 入力信号はインターフェイスをリセットします。

信号名	入力 / 出力	説明
reconfig_address[13:0]	入力	リードおよびライトのワードアドレス。
reconfig_read	入力	この信号をアサートしてリード転送を要求する必要があります。
reconfig_write	入力	この信号をアサートしてライト転送を要求する必要があります。
reconfig_writedata[31:0]	入力	ライトデータ
reconfig_readdata[31:0]	出力	リードデータ

continued...

信号名	入力 / 出力	説明
		reconfig_readdata[31:0]のデータは、reconfig_readがアサートされ、reconfig_waitrequest がディアサートされたクロックサイクルに続く reconfig_clk の立ち上がりエッジで有効です。
reconfig_waitrequest	出力	IP コアが準備できていないことを表示します。reconfig_waitrequest がアサートされている間は、入力信号の値を保持する必要があります。reconfig_readdata[31:0]のデータは、reconfig_waitrequest がアサートされている間は無効です。

関連情報

- 30 ページの [トランシーバー・リコンフィグレーション・インターフェイス](#)
- 36 ページの [テスト機能](#)
- [Avalon Interface Specifications](#)
タイミング図を含む Avalon-MM プロトコルについての詳細は、[Avalon メモリーマップド・インターフェイス](#)の章を参照してください。

4.8 外部 PLL へのインターフェイス上の信号

表 18. HMC コントローラー IP コアの外部 PLL インターフェイスの信号

HMC コントローラー IP コアは、各 HMC コントローラー IP コアのレーンに TX PLL IP コアを生成し、接続する必要があります。

信号名	入力 / 出力	説明
tx_bonding_clocks[95:0]	入力	個々のトランシーバー・チャンネルのクロック。各トランシーバー・チャンネルへの入力クロックは 6 ビットです。 この入力バスをトランシーバー TX PLL IP コアに接続する必要があります。レーンごとのデータレートの半分である出力周波数を指定するには、外部 TX PLL IP コアをパラメータ化する必要があります。HMC コントローラー IP コアの 10 Gbps レーンレートの場合は TX PLL IP コアの出力周波数は 5 GHz、12.5 Gbps レーンレートの場合は TX PLL IP コアの出力周波数は 6.25 GHz でなければなりません。また、15 Gbps レーンレートの場合、TX PLL IP コアの出力周波数は 7.5 GHz でなければなりません。
pll_locked	入力	外部 TX PLL からの PLL ロック表示。ユーザーロジックは、この入力信号を外部 TX PLL からの pll_locked インジケータで駆動する必要があります。 core_clk は、pll_locked がアサートされた後のみ安定します。IP コアは、core_rst_n 信号をディアサートして、core_clk が安定したことを示します。
pll_cal_busy	入力	外部 TX PLL からの PLL-busy 表示。アサートされると、PLL キャリブレーションが進行中であることを示します。

関連情報

- 21 ページの [外部 PLL の追加](#)
外部 トランシーバー PLL IP コアを生成し、HMC コントローラー IP コアに接続する方法を説明します。



5 HMC コントローラー IP コアのレジスターマップ

HMC コントローラー IP コアの内部レジスターは 32 ビット幅であり、*Avalon Interface Specifications* に準拠した Avalon-MM インターフェイスのコントロールおよびステータス・レジスター・インターフェイスを使用してアクセスできます。

これらのレジスターはすべて 32 ビット幅で、アドレス値は 16 進数で表示されます。レジスターは 32 ビット (4 バイト) 単位でのみアクセスできます。したがって、レジスターのアドレッシングは 4 の単位で増分されます。

Reserved または未定義の位置への書き込みアクセスは無効です。Reserved または未定義の位置への読み取りアクセスは未定義の結果を返します。

表 19. レジスター・アクセスコード

レジスタービットのタイプの記述に使用されるアクセスコードをリストしています。

コード	内容
RW	読み取り / 書き込み
RO	読み取り専用
RW1C	読み取り / クリアーへの 1 の書き込み
RTC	クリアーへの読み取り
WO	書き込み専用

表 20. コントロールおよびステータスレジスターのマップ

オフセット	レジスター名	追加情報の位置
0x00	Reserved	
0x04	CONTROL	コントロール・レジスター
0x08	XCVR_STATUS	トランシーバー・ステータス・レジスター
0x0C	LANE_STATUS	レーン・ステータスレジスター
0x10	LINK_STATUS	リンク・ステータスレジスター
0x14	ERROR_RESPONSE_CAPTURE	エラー応答レジスター
0x18	Reserved	
0x1C	LIMIT_OUTSTANDING_PACKETS	未解決応答パケットの FLIT の数の制限
0x20	INTERRUPT_STATUS	割り込みレジスター
0x24	INTERRUPT_ENABLE	
0x28	GLOBAL_INTERRUPT_ENABLE	
0x2C	Reserved	

continued...

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済

オフセット	レジスター名	追加情報の位置
0x30	LOCAL_ERROR_COUNT	スタティック・レジスター
0x34	REMOTE_ERROR_COUNT	
0x38	RETRY_BUFFER_ECC_COUNT	
0x3C	RESPONSE_QUEUE_ECC_COUNT	

5.1 CONTROL レジスター

表 21. HMC コントローラー IP コアのオフセット 0x04 にある CONTROL レジスター

ビット	フィールド名	タイプ	リセットの値	説明
RCLK[31..18]	Reserved	RO	0	
17	P_RST_N	RW	0x1	HMC のソフトウェア制御のリセット。IP コアは、hmc_p_rst_n 出力ポート上のこのフィールドの値を駆動し、この出力ポートは P_RST_N リセット信号に接続する必要があります。 IP コアの v15.0 との下位互換性のために、IP コアは rst_n 入力信号がアサートされている間はこの出力信号を強制的に Low にし、rst_n がデアサートされるとそれを発生させます。IP コアは値 1 に設定された後、出力信号は P_RST_N レジスターフィールドから駆動されます。 I
16	TXPS	RW	0x1	パワー・マネジメント・フィールド。IP コアは hmc_lxtps 出力ポートでこのフィールドの値を駆動し、HMC LxRXPS 入力ポートに接続されます。 IP コアの v15.0 との下位互換性のために、IP コアはこの出力信号を High にし、rst_n 入力信号がアサートされている間はこの出力信号を強制的に rst_n がデアサートされるとそれを発生させます。その後、出力信号は TXPS レジスターフィールドから駆動されます。
15:10	Reserved	RO	0x00	
9	ForceRXError	WO	0x0	値 1 をこのレジスターフィールドに書き込むと、HMC コントローラー IP コアを強制し、入力ストリームのエラーを検出し、StartRetry 要求を HMC デバイスに送信します。このビットはセルフクリアーです。
8	CRCErrortInje ct	WO	0x0	値 1 をこのレジスターフィールドに書き込むと、次の要求パケットの CRC で 1 ビット・エラーを注入します。このビットはセルフクリアーです。
RCLK[7..3]	Reserved	RO	0x00	
2	SoftReset	WO	0x0	このレジスターフィールドに 1 の値を書き込むと、レジスターとトランシーバーの送信側を除き、HMC コントローラー IP コアのすべての部分がリセットされます。このビットはセルフクリアーです。このビットを読み込むと常に 0 の値が返されます。
1	ClearFatalEr ror	WO	0x0	Retry State Machine (RSM) が致命的なエラー状態 (RetryFatalError) の場合、このレジスターフィールドに 1 の値を書き込むと RSM は通常の動作を再開します。RetryFatalError が生じると、RSM は停止して外部の訂正アクションを待機します。このレジスターフィールドに 1 の値を書き込むと、RSM が強制的に継続されます。このビットはセルフクリアーです。RSM ステートに影響を与えるかどうかをクリアーします。このビットを読み込むと常に 0 の値が返されます。
0	Retrain	WO	0x0	このレジスターフィールドに 1 の値を書き込むと、HMC コントローラー IP コアはリンク初期化シーケンスを再開します。このビットはセルフクリアーです。このビットを読み込むと常に 0 の値が返されます。

関連情報

- 36 ページの [テスト機能](#)



- 31 ページの 初期化とリセット

5.2 XCVR_STATUS レジスター

表 22. HMC コントローラー IP コアのオフセット 0x08 にある XCVR_STATUS レジスター

トランシーバー・チャンネル順での HMC リンクの個々のトランシーバー状態

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..16]	Reserved	RO	0x0001	
RCLK[15..0]	CDR Lock	RO	0x0000	各ビットが、対応するトランシーバー・チャンネルが受信データにロックされている CDR かどうかを表示します。

5.3 LANE_STATUS レジスター

表 23. HMC コントローラー IP コアのオフセット 0x0C にある LANE_STATUS レジスター

トランシーバー・チャンネル順での HMC リンクの個別のトランシーバー状態

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..16]	WordLock	RO	0x00	各ビットが対応する HMC リンクのトランシーバー・チャンネルが TS1 ワード境界にロックされているかどうかを表示します。
RCLK[15..0]	DescramSync	RO	0x00	各ビットが対応するトランシーバー・チャンネルのデスクランブラーが受信データに同期しているかどうかを示します。

5.4 LINK_STATUS レジスター

表 24. HMC コントローラー IP コアのオフセット 0x10 での LINK_STATUS レジスター

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..17]	Reserved	RO	0x0000	
16	RXPS	RO	0x0	デバイスからの LxTXPS 出力信号に接続される必要がある hmc_lxrtps 入力信号のレベル。
RCLK[15..9]	Reserved	RO	0x00	
8	LanesAligned	RO	0x0	受信データがすべてのレーンでアライメントされているかを表示します。
RCLK[7..6]	Reserved	RO	0x0	
RCLK[5..0]	InitializationState	RO	0x01	リンク初期化の現在の状態を表示します。このフィールドには次の有効な値があります。 <ul style="list-style-type: none"> • 6'b100000: アクティブ • 6'b010000: トランザクション初期化 (TRET を待機) • 6'b001000: ワード同期 (TS1 を送信) • 6'b000100: スランブラー同期 (NULL を送信) • 6'b000010: HMC コンフィグレーション (外部 I²C マスターまたは JTAG マスターモジュールによる) • 6'b000001: リセット

関連情報

31 ページの [初期化とリセット](#)

HMC コントローラー IP コア初期化中の InitializationState フィールドの動作について説明します。

5.5 ERROR_RESPONSE レジスター

表 25. HMC コントローラー IP コアのオフセット 0x14 にある ERROR_RESPONSE レジスター

HMC コントローラー IP コアは HMC リンクで受信するエラー応答の ERRSTAT および CUB フィールドを格納します。IP コアはこれらのフィールドを内部エラー応答キュー (FIFO バッファー) に格納します。アプリケーションは ERROR_RESPONSE レジスターを読み取り、このキューからの各エラー応答パケットの関連情報を読み出すことができます。レジスターからの読み取りにより、キューが進みます。

HMC コントローラー IP コアがゼロ以外の ERRSTAT フィールドで受信した読み取り、書き込み、または MODE 応答パケットは、このキューまたはレジスターに配線されません。代わりに、dp_rsp_error がアサートされとともにデータバスの応答インターフェイスに送られます。

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..17]	Reserved	RO	0x0000	
16	Valid	RO	0x0	レジスターの CUB および ERRSTAT フィールドが有効値を保持していることを表示します。エラー応答キューが空の場合、CUB および ERRSTAT フィールドは無効であり、Valid ビットは 0 の値を持っています。 Valid ビットをポーリングして応答パケットが処理の待機中かどうかを判断するか、INTERRUPT_ENABLE レジスターの RX Error Response 割り込みを有効にできません。
15:11	Reserved	RO	0x00	
RCLK[10..8]	CUB	RO	0x0	エラー応答パケットの TAG フィールドから抽出した CUB ID。
7	Reserved	RO	0x0	
RCLK[6..0]	ERRSTAT	RO	0x00	エラー応答パケットから抽出した ERRSTAT 値。

関連情報

[Hybrid Memory Cube Specification 1.1](#)

ERRSTAT 応答パケットフィールドのエンコーディングに関する情報は HMC specification の Table 16 でご確認ください。

5.6 LIMIT_OUTSTANDING_PACKETS レジスター



表 26. HMC コントローラー IP コアのオフセット 0x1C にある LIMIT_OUTSTANDING_PACKETS レジスター

ビット	フィールド名	タイプ	リセット値	説明
31	Enable	RW	0x0	このレジスターフィールドに 1 の値を書き込むと、Limit Outstanding FLIT 機能がオンになります。この機能がオンになっている場合、IP コアは未処理の応答パケットの FLIT の数を MaxRspPktFlit フィールドで指定されたしきい値に制限します。この機能を有効にすると、予期される応答がしきい値を超えた場合、IP コアが HMC に要求を送信しない効果があります。 機能がオンであるかオフであるかにかかわらず、IP コアはリトライ・バッファースペースとトークンが使用可能な場合にのみ、HMC に要求パケットを送信します。クライアントは、クライアントがいつでも予期された応答を受信できる場合にのみ、データバス要求インターフェイスで要求を送信する責任があります。IP コアは、dp<n>_req_ready 信号をデアサートすることで、データバス要求インターフェイスを背圧することができます。Limit Outstanding FLIT 機能または Response Open Loop モード機能で IP コアが HMC に要求パケットを送信できない場合、IP コアは dp<n>_req_ready 信号をデアサートします。
RCLK[30..10]	Reserved	RO	0x000000	
RCLK[9..2]	MaxRspPktFlit	RW	0x00	Limit Outstanding FLIT 機能がオンの場合、最大 FLIT の数を指定します。しきい値は、このレジスターフィールドの値の 4 倍です。 このレジスターの一般的な値は 150 です。しかしながら、デザインのトラフィック・パターンとシステム構成を考慮し、スループットやレイテンシーの向上のためにこの値を調整することを推奨します。値を小さくするとレイテンシーが低下しますが、RX スループットも低下します。受け入れ可能なスループットをもたらす最低値を使用してください。
RCLK[1..0]	Reserved	RO	0x0	

関連情報

34 ページの [フロー制御](#)

5.7 割り込み関連のレジスター

HMC コントローラー IP コアは 3 つの割り込みに関連するレジスターがあります。

- INTERRUPT_STATUS: レジスタービットは個別の割り込みソースステータスを報告します。
- INTERRUPT_ENABLE: GLOBAL_INTERRUPT_ENABLE レジスターがこの機能をオフにしない限り、レジスタービットは INTERRUPT_STATUS レジスターの対応する割り込みを個別に有効にし、IP コアの csr_irq 出力信号のアサートをトリガーします。
- GLOBAL_INTERRUPT_ENABLE: レジスターを使用すると、すべての割り込み応答を無効にするか、INTERRUPT_ENABLE レジスターに表示されている割り込みソースを有効にできます。

表 27. HMC コントローラー IP コアのオフセット 0x20 にある INTERRUPT_STATUS レジスター

割り込みをクリアするには、割り込みビットに 1 の値を書き込みます。

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..16]	Reserved	RO	0x0000	
15	Response Queue Uncorrectable ECC Error	W1C	0x0	IP コアは応答キューメモリーで訂正不可能な ECC エラーを検出すると、このビットを設定します。IP コアはパラメーター・エディターの Enable M20K ECC support をオンにした場合にのみ、このようなエラーを検出できます。

continued...



ビット	フィールド名	タイプ	リセット値	説明
14	Response Queue ECC Error	W1C	0x0	IP コアは応答キューメモリーで訂正可能な ECC エラーを検出すると、このビットを設定します。IP コアはパラメーター・エディターの Enable M20K ECC support をオンにした場合にのみ、このようなエラーを検出できます。
13	FERR_N	W1C	0x0	IP コアは HMC デバイスが FERR_N ピンがアクティブ Low にアサートされることで致命的なエラーを表示すると、このビットを設定します。IP コアの hmc_ferr_n 入力信号を HMC デバイスの FERR_N 出力信号に接続する必要があります。
12	Retry Buffer Uncorrectable ECC Error	W1C	0x0	IP コアはリトライ・バッファー・メモリーで訂正不可能な ECC エラーを検出すると、この割り込みビットを設定します。IP コアはパラメーター・エディターの Enable M20K ECC support をオンにした場合にのみ、このようなエラーを検出できます。
11	Retry Buffer ECC Error	W1C	0x0	IP コアはリトライ・バッファー・メモリーで訂正可能な ECC エラーを検出すると、この割り込みビットを設定します。この場合、IP コアは自動的に ECC エラーを訂正します。IP コアはパラメーター・エディターの Enable M20K ECC support をオンにした場合にのみ、このようなエラーを検出できます。
10	Reserved	RO	0x0	
9	No More Tokens	W1C	0x0	IP コアはトークンが実行されない場合にこの割り込みビットを設定します。トークンは HMC デバイスで使用可能なバッファースペースを表します。IP コアはトークンを保持していませんが、トークンベースのフロー制御要件ごとの追加の要求は送信されません。この状況はエラー状態ではありませんが、パフォーマンスが低下している可能性があります。しかしながら、割り込みビットと同様に、IP コアは csr_irq 信号をアサートします (グローバル割り込みイネーブルレジスター・ビットが設定されていると仮定します)。 このビットは IP コアがリセット解除されたときに値 0 を有します。リンクの初期化後、HMC デバイスはそのバッファー容量を一連の TRET パケットで通信します。IP コアが最初の TRET パケットを受信すると、No More Tokens レジスターフィールドの更新を開始します。
8	Retry Buffer Full	W1C	0x0	IP コアはリトライバッファーが満量になると、この割り込みビットを設定します。リトライバッファーが満量になった場合、IP コアは追加のリードまたはライト要求を送信しません。この状況はエラー状態ではありませんが、パフォーマンスが低下している可能性があります。
7	Reserved	RO	0x0	
6	RX Error Response Overflow	W1C	0x0	IP コアは、ERROR_RESPONSE レジスターから読み取られる前に過度なエラー応答パケットを受信すると、この割り込みビットを設定します。オーバーフローが発生した場合、IP コアはエラー応答キューでスペースが再び使用可能になるまで、受信するエラー応答パケットを排除します。
5	RX Error Response	W1C	0x0	IP コアがエラー応答パケットを受信すると、この割り込みビットを設定します。
4	Fatal Error	W1C	0x0	IP コアは未成功に対し 3 回以上の連続した再試行が実行されると、この割り込みビットを設定します。
3	Remote Error	W1C	0x0	IP コアは HMC デバイスがエラーを検出したことを示す有効な IRTRY (SartRetry) シーケンスを受信すると、この割り込みビットを設定します。
2	SEQ Error	W1C	0x0	IP コアは、受信した前のパケットの SEQ フィールド値から +1 の増分ではない SEQ フィールド値を持つパケットを受信すると、この割り込みビットを設定します。
1	LNG/DLN Error	W1C	0x0	IP コアは、LNG (パケット長) および DLN (重複長) フィールドの等しくないまたは無効な値のパケットを受信すると、この割り込みビットを設定します。
0	CRC Error	W1C	0x0	IP コアは、受信したパケットの CRC のエラーを検出すると、この割り込みビットを 1 の値に設定します。

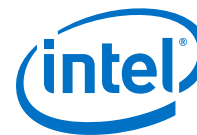


表 28. HMC コントローラー IP コアのオフセット 0x24 にある INTERRUPT_ENABLE レジスター

このレジスターの各ビットはオフセット 0x20 にある INTERRUPT_STATUS レジスターの対応する割り込みを有効にします。各レジスタービットでは次が実行されます。

- ビットが 0 の値を有する場合、割り込みをディスエーブルする。
- ビットが 1 の値を有し、オフセット 0x28 にある GLOBAL_INTERRUPT_ENABLE レジスターの GlobalEnable ビットが 1 の値を有する場合、割り込みをイネーブルする。

ビット	フィールド名	タイプ	リセット値	説明
RCLK[31..16]	Reserved	RO	0x0000	
15	Response Queue Uncorrectable ECC Error Enable	RW	0x0	Response Queue Uncorrectable ECC Error 割り込みを有効にします。
14	Response Queue ECC Error Enable	RW	0x0	Response Queue ECC Error 割り込みを有効にします。
13	FERR_N Enable	RW	0x0	FERR_N 割り込みを有効にします。
12	Retry Buffer Uncorrectable ECC Error Enable	RW	0x0	Retry Buffer Uncorrectable ECC Error 割り込みを有効にします。
11	Retry Buffer ECC Error Enable	RW	0x0	Retry Buffer ECC Error 割り込みを有効にします。
10	Reserved	RO	0x0	
9	No More Tokens Enable	RW	0x0	No More Tokens 割り込みを有効にします。
8	Retry Buffer Full Enable	RW	0x0	Retry Buffer Full 割り込みを有効にします。
7	Reserved	RO	0x0	
6	RX Error Response Overflow Enable	RW	0x0	RX Error Response Overflow 割り込みを有効にします。
5	RX Error Response Enable	RW	0x0	RX Error Response 割り込みを有効にします。
4	Fatal Error Enable	RW	0x0	Fatal Error 割り込みを有効にします。
3	Remote Error Enable	RW	0x0	Remote Error 割り込みを有効にします。
2	SEQ Error Enable	RW	0x0	SEQ Error 割り込みを有効にします。
1	LNG/DLN Error Enable	RW	0x0	LNG/DLN Error 割り込みを有効にします。
0	CRC Error Enable	RW	0x0	CRC Error 割り込みを有効にします。

表 29. HMC コントローラー IP コアのオフセット 0x28 にある GLOBAL_INTERRUPT_ENABLE レジスター

INTERRUPT_ENABLE レジスターをゲートします。

ビット	フィールド名	タイプ	リセット値	説明
31:1	Reserved	RO	0x00000000	
0	GlobalEnable	RW	0x0	このレジスターフィールドに 0 の値を書き込むと、csr_irq 出力信号のアサートからのすべての割り込みソースがディスエーブルされます。 このレジスターフィールドに 1 の値を書き込むことで、IP コアはオフセット 0x24 にある INTERRUPT_ENABLE レジスターでイネーブルにされた割り込みソースに従って csr_irq 出力信号をアサートできます。割り込みソースは GlobalEnable レジスターフィールドと関連する INTERRUPT_ENABLE レジスターフィールドの両方が 1 の値を有する場合にのみ、IP コアが csr_irq 出力信号をアサートします。

5.8 エラーおよびリトライ・スタティック・レジスター

HMC コントローラー IP コアには 4 つのスタティック・レジスターがあります。3 つのカウンターフィールドはすべて RC (クリアーへの読み出し) タイプです。

表 30. HMC コントローラー IP コアのオフセット 0x30 にある LOCAL_ERROR_COUNT レジスター

ビット	フィールド名	タイプ	リセット値	説明
31:16	Reserved	RO	0x0000	
15:0	Local Count	RC	0x0000	CRC、SEQ、または長さエラーの受信パケットのカウンタ。カウンターは 0xFFFF で飽和します。 このレジスタークリアーを読み出すと、Local Count フィールドをクリアーします。

表 31. HMC コントローラー IP コアのオフセット 0x34 にある REMOTE_ERROR_COUNT レジスター

ビット	フィールド名	タイプ	リセット出の値	説明
31:16	Reserved	RO	0x0000	
15:0	Error Count	RC	0x0000	HMC コントローラー IP コアが出力エラー回復プロセスを開始し、パケットを再送信した回数。この数は外部 HMC デバイスによりエラー検出された数を示します。このカウンターは 0xFFFF で飽和します。 このレジスタークリアーを読み出すと、Error Count フィールドをクリアーします。

表 32. HMC コントローラー IP コアのオフセット 0x38 にある RETRY_BUFFER_ECC_COUNT レジスター

ビット	フィールド名	タイプ	リセット値	説明
31:24	Reserved	RO	0x00	
23:16	Uncorrectable Count	RC	0x00	IP コアがリトライ・バッファ・メモリーで検出した訂正不可能な ECC エラーの数。このカウンターは 0xFF で飽和します。 このフィールドはパラメーター・エディターで Enable M20K ECC support をオンにしない限り、0 の値を保持します。

continued...



ビット	フィールド名	タイプ	リセット値	説明
				このレジスターはすべての Uncorrectable Count フィールドを読み出すと、フィールドをクリアします。
15:8	Reserved	RO	0x00	
7:0	Correctable Count	RC	0x00	IP コアがリトライ・バッファメモリーで検出した (訂正済みの) 訂正可能な ECC エラーの数。このカウンタは 0xFF で飽和します。 このフィールドは、パラメーター・エディターで Enable M20K ECC support をオンにしない限り、0 の値を保持します。 このレジスターはすべての Correctable Count フィールドを読み出すと、フィールドをクリアします。

表 33. HMC コントローラー IP コアのオフセット 0x3C にある RESPONSE_QUEUE_ECC_COUNT レジスター

ビット	フィールド名	タイプ	リセット値	説明
31:24	Reserved	RO	0x00	
23:16	Uncorrectable Count	RC	0x00	IP コアが応答キューメモリーで検出した訂正不可能な ECC エラーの数。このカウンタは 0xFF で飽和します。 このフィールドは、パラメーター・エディターで Enable M20K ECC support をオンにしない限り、0 の値を保持します。 このレジスターは Uncorrectable Count フィールドを含めたすべてのフィールドを読み出します。
15:8	Reserved	RO	0x00	
7:0	Correctable Count	RC	0x00	IP コアが応答キューメモリーで検出した (訂正済みの) 訂正可能な ECC エラーの数。このカウンタは 0xFF で飽和します。 このフィールドは、パラメーター・エディターで Enable M20K ECC support をオンにしない限り、0 の値を保持します。 このレジスターはすべての Correctable Count フィールドを読み出すと、フィールドをクリアします。

関連情報

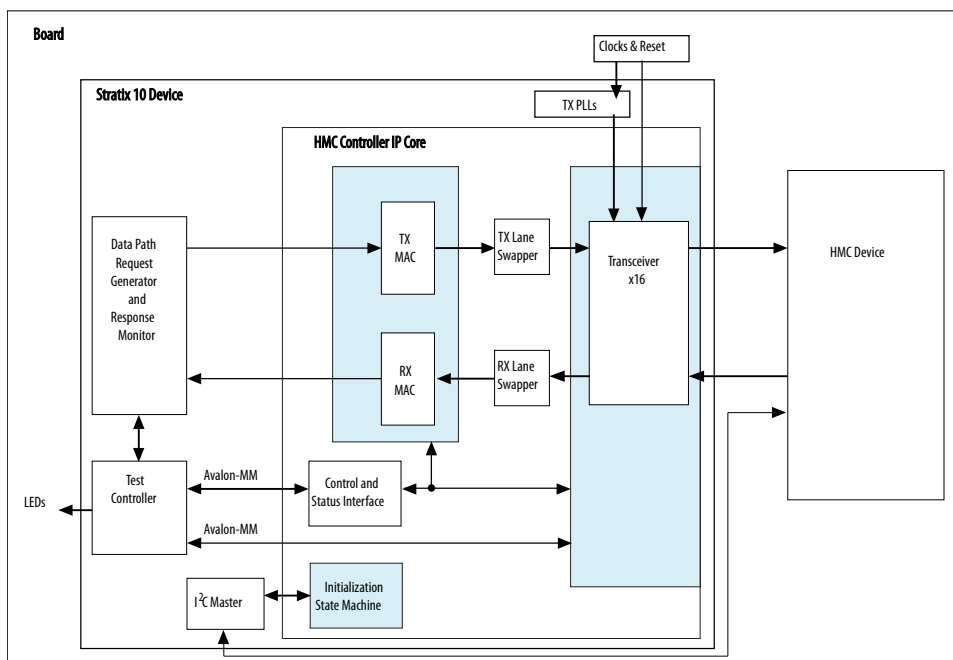
34 ページの [M20K ECC サポート](#)

6 HMC コントローラー IP コア Stratix 10 のデザイン例

インテルは Stratix 10 デバイスをターゲットとする HMC コントローラー IP コアを使用したシミュレーションおよびコンパイル準備のデザイン例を提供しています。このデザインはタイミング収束が保証されていません。Arria 10 パージョンのデザイン例については、*Hybrid Memory Cube Controller Design Example User Guide* を参照してください。現在、この IP コアの Stratix 10 パージョンはボードをサポートしていませんが、シミュレーションおよびコンパイルに使用できます。

注意: このデザイン例はシミュレーションおよびコンパイルに使用できますが、タイミング収束は保証されていません。また、IP コアの Quartus Prime Pro – Stratix 10 Edition Beta パージョンは、Mentor Graphics Modelsim シミュレーターと Synopsys VCS シミュレーターのみをサポートしています。

図 -20: HMC コントローラー IP コアのデザイン例における上位レベルのブロック図



関連情報

- [「すべての開発キット」に関するウェブページ](#)
デザイン例を対象とするインテル開発キットについての情報です。
- [Quartus Prime Standard Edition Handbook, Volume 3: Verification](#)
インテル デバイスのプログラミングについての詳細は、「Programming Intel Devices」の章を参照してください。
- [Arria 10 Hybrid Memory Cube Controller Design Example User Guide](#)



A HMC コントローラー IP コア・ユーザーガイド・アーカイブ

IP コアのバージョンが記載されていない場合には、以前の IP コアバージョン向けのユーザーガイドが適用されます。

IP コアバージョン	ユーザーガイド
16.0	ハイブリッド・メモリー・キューブ・コントローラー IP コア・ユーザーガイド
15.0	ハイブリッド・メモリー・キューブ・コントローラー IP コア・ユーザーガイド

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



B HMC コントローラー IP ユーザーガイドの資料改訂履歴

B.1 HMC コントローラー IP ユーザーガイド改訂履歴

表 34. 改訂履歴

HMC コントローラー IP コア におけるユーザーガイドの新機能および変更箇所の要約です。

日付	ACDS バージョン	変更内容
2016 年 8 月 8 日	Quartus Prime Pro – Stratix 10 Edition Beta	<p>Quartus Prime Pro – Stratix 10 Edition Beta ソフトウェア・リリースの更新：</p> <ul style="list-style-type: none"> Stratix 10 デバイスのサポートを追加。 IP コアの Arria 10 バリエーションの記述を削除。このリリースは Stratix 10 デバイスのみをサポート。 全幅バリエーションの記述を削除。半幅バリエーションは Stratix 10 デバイスファミリーのみをサポート。 このリリースのサポートされるシミュレーションおよびデザイン例でのいくつかの差異を注記。HMC コントローラー IP コア Stratix 10 のデザイン例 60 ページの を参照。 レジスター名を変更。HMC コントローラー IP コアのレジスターマップ 51 ページの を参照。 <ul style="list-style-type: none"> レジスター名を LIMIT_OUTSTANDING_PACKET から LIMIT_OUTSTANDING_PACKETS へ変更。LIMIT_OUTSTANDING_PACKETS レジスター 54 ページの を参照。 レジスター名を RESPONSE_BUFFER_ECC_COUNT から RESPONSE_QUEUE_ECC_COUNT へ変更。エラーおよびリトライ・スタティック・レジスター 58 ページの を参照。 このリリースでは未サポートだが、IP コア・パラメーター・エディターでまだ確認できる Lanes および Enable ADME and Optional Reconfiguration Logic IP コアパラメーターを明記。HMC コントローラー IP コアのパラメーター 13 ページの を参照。 このリリースで、トランシーバー・リコンフィグレーション・インターフェイスが機能しないことを明記。ただし、適切なクロック周波数で reconfig_clk 入力を駆動する必要がある。トランシーバー・リコンフィグレーション・インターフェイス 30 ページの、トランシーバー・リコンフィグレーション信号 49 ページの および上位レベルのブロック図 27 ページの を参照。 rst_n および reconfig_reset 入力信号が非同期であり、IP コアがリセット要求のキャプチャーに十分な長さでサポートされる必要があることを明記。クロック信号およびリセット信号 48 ページの を参照。 隣接する 3 ビット・エラー検出 (Arria 10 デバイス向け) から隣接する 3 ビット・エラー訂正 (Stratix 10 デバイス向け) に M20K ECC 機能の説明を更新。HMC コントローラー IP コアのパラメーター 13 ページの および M20K ECC サポート 34 ページの を参照。 各 HMC コントローラー IP コアをシングル HMC デバイスのリンクに接続することを明記。HMC デバイスの複数のリンクを接続するには、HMC コントローラー IP コアの複数のインスタンスを初期化する必要がある。HMC コントローラー IP コアでサポートされる機能 5 ページの を参照。

continued...

Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Altera、ARRIA、CYCLONE、ENPIRION、MAX、NIOS、QUARTUS および STRATIX の名称およびロゴは、アメリカ合衆国および/またはその他の国における Intel Corporation の商標です。インテルは FPGA 製品および半導体製品の性能がインテルの標準保証に準拠することを保証しますが、インテル製品およびサービスは、予告なく変更される場合があります。インテルが書面にて明示的に同意する場合を除き、インテルはここに記載されたアプリケーション、または、いかなる情報、製品、またはサービスの使用によって生じるいっさいの責任を負いません。インテル製品の顧客は、製品またはサービスを購入する前、および、公開済みの情報を信頼する前には、デバイスの仕様を最新のバージョンにしておくことをお勧めします。

*その他の社名、製品名などは、一般に各社の表示、商標または登録商標です。

ISO
9001:2008
登録済



日付	ACDS バージョン	変更内容
		<ul style="list-style-type: none"> 特定のペイロードサイズが半幅のバリエーションでのみ使用可能である誤表示を削除。アプリケーション応答インターフェイス 41 ページの を参照。 項「クロッキングおよびリセット構造」から「クロッキング構造」へ変更。この項はリセット情報には含まれていない。クロッキング構造 30 ページの を参照。 誤字およびマイナーエラーの修正。
2016.05.02	16.0	<ul style="list-style-type: none"> Quartus Prime ソフトウェアの v16.0 リリースの更新。 全幅のバリエーションで複数の (2, 3, または 4) データ・インターフェイスのサポートを追加。 <ul style="list-style-type: none"> ポート (1, 2, 3, または 4) の数を指定するための新しい Ports パラメーターを追加。HMC コントローラー IP コアのパラメーター 13 ページの を参照。 新しい信号を追加インターフェイス向けに追加。データバス・インターフェイスでの信号名は、1 つ以上のポートの場合は <code>dp<n>_req_*</code> および <code>dp<n>_rsp_*</code> で、<code><n></code> はポートナンバー (0, 1, 2, 3) である。アプリケーション要求インターフェイス 37 ページ の および アプリケーション応答インターフェイス 41 ページの を参照。 IP コアが対応するデータ要求インターフェイスで元の要求を受信した順に各データ応答インターフェイスで応答を返すことを指定するように、全幅のバリエーションでの新しい Response re-ordering パラメーターを追加。この新オプションをオンにすると、IP コアは内部にタグ管理を実装し、タグはデータ・インターフェイス上で可視化できない。HMC コントローラー IP コアでサポートされる機能 5 ページ の および HMC コントローラー IP コアのパラメーター 13 ページの を参照。 全幅のバリエーションの 2 のべき乗のペイロードサイズの READ、WRITE、および ポステッド WRITE トランザクションの追加によるサポートするトランザクションのリストを拡張。現在、全幅のバリエーションは半幅のバリエーションをサポートするすべてのトランザクション・タイプをサポートする。HMC コントローラー IP コアがサポートする HMC トランザクションのタイプ 6 ページの を参照。 パラメーター名を Enable Altera Debug Master Endpoint (ADME) から Enable ADME and Optional Reconfiguration Logic へ変更。パラメーターが設定する PHY デバッグ機能のリストを追加。HMC コントローラー IP コアのパラメーター 13 ページの を参照。 TX PLL xN bonding モードをサポートする周波数への CDR reference clock パラメーターの許容値リストを制限。 新しいデータバス応答インターフェイス信号 <code>dp<n>_rsp_errstat[6:0]</code> を追加。この信号は外部 HMC からの応答パケットの ERRSTAT フィールドの値を保持する。アプリケーション応答インターフェイス 41 ページの を参照。 <code>pll_powerdown</code> 出力信号を削除。外部 PLL の追加 21 ページ の および 外部 PLL へのインターフェイス上の信号 50 ページの を参照。 現在、IP コアがマルチサイクル・トランザクションの間に <code>dp<n>_rsp_error</code> の値を保持することを示すため、<code>dp<n>_rsp_error</code> 信号の説明を更新。以前はこの動作は保証されていない。アプリケーション応答インターフェイス 41 ページの を参照。 HMC 仕様の表 25 からのノンエラー応答コードのみを保持することを明記するため、<code>dp<n>_rsp_cmd</code> 信号の説明を修正。アプリケーション応答インターフェイス 41 ページの を参照。 リード応答の輻輳を緩和するために、全幅のバリエーションに新しい Limit Outstanding FLIT 機能を追加。機能の制御に新しい <code>LIMIT_OUTSTANDING_PACKET</code> レジスターを追加。フロー制御 34 ページ の および LIMIT_OUTSTANDING_PACKETS レジスター 54 ページの を参照。 リセット、リンク初期化、致命的なエラーの回復、およびパワー・マネージメントのソフトウェア制御を追加。CONTROL レジスターに次のフィールド追加。 <ul style="list-style-type: none"> <code>P_RST_N</code> in bit [17]: HMC の制御するソフトウェア <code>TXPS</code> in bit [16]: パワー・マネージメント・フィールド <code>SoftReset</code> in bit [2]: ソフトウェアが制御する IP コアのリセット <code>Retrain</code> in bit [0]: IP コアのリンク初期化のシーケンスの再スタート 初期化とリセット 31 ページ の および CONTROL レジスター 52 ページの を参照。 HMC インターフェイス信号 44 ページ の <code>hmc_1ltxps</code> 信号の説明で信号方向のリファレンスを修正。
2015年5月4日	15.0	初版