



# AN 802: インテル® Stratix® 10 SoC デバイスのデザイン・ガイドライン

インテル® Quartus® Prime 開発デザインスイートの更新情報: **18.1**



更新情報

フィードバック

AN-802 | 2019.04.17

最新版をウェブからダウンロード: [PDF](#) | [HTML](#)

## 目次

<b>1. インテル® Stratix® 10 SoC デバイスのデザイン・ガイドライン概要</b> .....	<b>4</b>
1.1. Stratix 10 SoC デバイスのデザイン・ガイドライン概要の改訂履歴 .....	4
<b>2. Stratix 10 SoC FPGA のボード・デザイン・ガイドライン</b> .....	<b>5</b>
2.1. HPS のクロックおよびリセットのデザインにおける考慮事項.....	5
2.1.1. HPS クロックのプランニング.....	5
2.1.2. 早期ピン・プランニングおよび I/O 割り当ての解析.....	6
2.1.3. HPS クロック、リセット、PoR のピン機能と接続.....	6
2.1.4. リモート・システム・アップデート (RSU) 機能に向けた Direct-to-Factory ピンのサポート....	6
2.1.5. 内部クロック.....	7
2.1.6. HPS ペリフェラルのリセット管理.....	7
2.2. デバイス I/O を HPS ペリフェラルおよびメモリーに接続するためのデザインにおける考慮事項.....	8
2.2.1. HPS ピンの多重化におけるデザインの考慮事項.....	9
2.2.2. HPS I/O の設定: 制約とドライブ強度.....	9
2.3. HPS インターフェイスにおけるデザイン・ガイドライン.....	10
2.3.1. HPS EMAC PHY インターフェイス.....	10
2.3.2. USB インターフェイスのデザイン・ガイドライン.....	18
2.3.3. SD/MMC および eMMC カード・インターフェイスのデザイン・ガイドライン.....	22
2.3.4. フラッシュ・インターフェイスのデザイン・ガイドライン.....	22
2.3.5. UART インターフェイスのデザイン・ガイドライン.....	23
2.3.6. I <sup>2</sup> C インターフェイスのデザイン・ガイドライン.....	24
2.4. HPS EMIF デザインの考慮事項.....	25
2.4.1. HPS を SDRAM に接続するための考慮事項.....	27
2.4.2. HPS SDRAM I/O の位置.....	28
2.5. HPS メモリーのデバッグ.....	30
2.6. HPS のバウンダリー・スキャン.....	31
2.7. エンベデッド・ソフトウェアのデバッグとトレース.....	31
2.8. インテル Stratix 10 SoC FPGA のボード・デザイン・ガイドライン改訂履歴 .....	32
<b>3. Stratix 10 SoC FPGA の FPGA との接続</b> .....	<b>33</b>
3.1. HPS メモリーマップド・インターフェイスの概要.....	33
3.1.1. HPS-to-FPGA ブリッジ.....	34
3.1.2. Lightweight HPS-to-FPGA ブリッジ.....	34
3.1.3. FPGA-to-HPS ブリッジ.....	35
3.1.4. FPGA-to-SDRAM ポート.....	35
3.1.5. インターフェイス帯域幅.....	35
3.2. 推奨されるシステムトポロジー.....	37
3.2.1. FPGA ファブリックへの HPS のアクセス.....	38
3.2.2. FPGA とデータを共有する MPU.....	38
3.2.3. FPGA からのキャッシュ可能およびキャッシュ不可能なデータアクセスの例.....	39
3.3. HPS-to-FPGA インターフェイス・デザインに推奨される開始点.....	43
3.4. FPGA アクセラレーターに向けたタイミング・クロージャック.....	43
3.5. ブリッジのコンフィグレーションおよび使用方法に関する情報.....	43
3.6. インテル Stratix 10 SoC FPGA の FPGA との接続の改訂履歴 .....	44



<b>4. Stratix 10 SoC FPGA のシステムに関する考慮事項</b> .....	<b>45</b>
4.1. タイミングに関する考慮事項.....	45
4.1.1. FPGA アクセラレーターに向けたタイミング・クロージャークロージャ.....	45
4.1.2. USB インターフェイスのデザイン・ガイドライン.....	46
4.2. 最大限のパフォーマンスの実現.....	49
4.3. システムレベルのキャッシュ・コヒーレンシー.....	49
4.4. Stratix 10 SoC FPGA のシステムに関する考慮事項の改訂履歴 .....	50
<b>5. インテル Stratix 10 SoC FPGA 向けエンベデッド・ソフトウェアのデザイン・ガイドライン</b> .....	<b>51</b>
5.1. 概要.....	51
5.2. ソフトウェア開発プラットフォームのコンポーネントの組み立て.....	51
5.3. ゴールデン・ハードウェア・リファレンス・デザイン (GHRD).....	52
5.4. アプリケーションのオペレーティング・システムの選択.....	52
5.4.1. Linux または RTOS の使用.....	52
5.4.2. ベアメタル・アプリケーションの開発.....	53
5.4.3. ベアメタル・フレームワークとしてのブートローダーの使用.....	54
5.4.4. 対称型および非対称型マルチプロセッシング・モード (SMP および AMP) .....	54
5.5. Linux に向けたソフトウェア開発プラットフォームの構築.....	55
5.5.1. Linux 向けゴールデン・システム・リファレンス・デザイン (GSRD).....	55
5.5.2. ソースコード管理における考慮事項.....	56
5.6. ベアメタル・アプリケーションに向けたソフトウェア開発プラットフォームの構築.....	56
5.7. パートナー OS または RTOS に向けたソフトウェア開発プラットフォームの構築.....	57
5.8. ブートローダー・ソフトウェアの選択.....	57
5.9. 開発、デバッグ、およびトレースに向けたソフトウェア・ツールの選択.....	58
5.9.1. ソフトウェア・ビルド・ツールの選択.....	59
5.9.2. ソフトウェア・デバッグ・ツールの選択.....	59
5.9.3. ソフトウェア・トレース・ツールの選択.....	60
5.10. ブートおよびコンフィグレーションにおける考慮事項.....	60
5.10.1. コンフィグレーション・ソース.....	60
5.10.2. コンフィグレーション・フラッシュ.....	60
5.10.3. コンフィグレーション・クロック.....	61
5.10.4. HPS ブートオプションの選択.....	61
5.10.5. HPS ブートソース.....	61
5.10.6. リモート・システム・アップデート (RSU).....	62
5.11. システムリセットにおける考慮事項.....	62
5.12. フラッシュの考慮事項.....	63
5.12.1. フラッシュのプログラミング方法.....	63
5.12.2. FPGA コンフィグレーションおよび HPS 大容量ストレージへのシングルフラッシュの使用..	64
5.13. エンベデッド・ソフトウェアのデバッグとトレース.....	64
5.14. インテル Stratix 10 SoC FPGA 向けエンベデッド・ソフトウェアのデザイン・ガイドライン改訂履歴 .....	65
<b>6. Stratix 10 SoC FPGA に関連する推奨リソース</b> .....	<b>66</b>
6.1. デバイスのドキュメント.....	66
6.2. ツールとソフトウェアのウェブページ.....	66
6.3. インテル Stratix 10 SoC FPGA に関連する推奨リソースの改訂履歴 .....	67

## 1. インテル® Stratix® 10 SoC デバイスのデザイン・ガイドライン概要

このデザイン・ガイドラインは、インテル® Stratix® 10 SoC FPGA デバイスを使用するデザインに向けた一連のガイドラインと推奨事項、および考慮すべき項目を提供することを目的としています。このガイドラインは、インテル Stratix 10 SoC FPGA デザイン、プラットフォーム・デザイナーのサブシステム・デザイン、ボードデザイン、ソフトウェア・アプリケーション・デザインのプランニングおよび初期設計段階において役立ちます。

このアプリケーション・ノートには、インテル Stratix 10 ハード・プロセッサ・システム (HPS) デバイスの詳細、機能、ハードウェアまたはソフトウェア・システムのデザインに関する情報がすべて含まれていないわけではありません。

インテル Stratix 10 HPS の機能および各ペリフェラルについての詳細は、*Intel Stratix 10 Hard Processor System Technical Reference Manual* を参照ください。

### 注意:

インテルでは、インテル® Quartus® Prime 開発ソフトウェア・プロ・エディションおよびインテル® SoC FPGA エンベデッド開発スイート (SoC EDS) プロ・エディションを使用し、インテル Stratix 10 SoC のデザインを開発することを推奨しています。インテル Quartus Prime 開発ソフトウェア・スタンダード・エディションおよびインテル SoC FPGA エンベデッド開発スイート・スタンダード・エディションは引き続き インテル Stratix 10 SoC ファミリーをメンテナンス・ベースでサポートしますが、今後の機能拡張はプロ・エディションでのみサポートされる予定です。インテル Quartus Prime 開発ソフトウェア・プロ・エディションで開発されたハードウェアは、インテル SoC FPGA エンベデッド開発スイート・プロ・エディションで開発されたソフトウェアのみをサポートします。

インテル Quartus Prime 開発ソフトウェア・スタンダード・エディションで開発されたハードウェアは、インテル SoC FPGA エンベデッド開発スイート・スタンダード・エディションで開発されたソフトウェアのみをサポートします。

### 関連情報

[Intel Stratix 10 Hard Processor System Technical Reference Manual](#)

### 1.1. Stratix 10 SoC デバイスのデザイン・ガイドライン概要の改訂履歴

表 1. Stratix 10 SoC デバイスのデザイン・ガイドライン概要の改訂履歴

ドキュメント・バージョン	変更内容
2019.04.17	メンテナンス・リリース
2019.03.19	メンテナンス・リリース
2018.12.24	メンテナンス・リリース
2018.09.24	メンテナンス・リリース
2018.05.07	メンテナンス・リリース
2018.03.01	メンテナンス・リリース
2017.11.06	初版

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  
9001:2015  
登録済



## 2. Stratix 10 SoC FPGA のボード・デザイン・ガイドライン

### 2.1. HPS のクロックおよびリセットのデザインにおける考慮事項

HPS の主なクロックおよびリセットソースは次のとおりです。

- HPS\_OSC\_CLK デバイス I/O ピン—HPS PLL の外部クロックソースです。HPS PLL は、MPU サブシステム、CCU、SMMU、L3 インターコネク、HPS ペリフェラルおよび HPS-to-FPGA ユーザークロックにクロックを生成します。
- nCONFIG デバイス I/O ピン—SoC デバイス全体のリセット入力で、FPGA をリコンフィグレーションし、HPS をコールドリセットします。
- HPS\_COLD\_nRESET デバイス I/O ピン—オプションのリセット入力で、HPS のみをコールドリセットし、双方向動作にコンフィグレーションされます。

**ガイドライン:** HPS\_COLD\_nRESET ピンは、オープンになっている任意の SDM I/O ピンにコンフィグレーションできます。

インテル Quartus® Prime で以下の作業を行います。

1. **Assignments > Device** の順にクリックします。
2. 「Device and Pin Options」ボタンをクリックします。
3. 「Configuration」タブに移動します。
4. 「Configuration Pin Options」ボタンをクリックします。
5. 「USE\_HPS\_COLD\_nRESET」のチェックボックスをクリックし、利用可能な SDM\_IO ピンを選択します。

詳細は、「HPS クロック、リセット、POR のピン機能と接続」の章を参照ください。

#### 2.1.1. HPS クロックのプランニング

HPS クロックのプランニングには、次のコンポーネントに対するクロックソースの選択と動作周波数の定義が含まれます。

- HPS PLL
- MPU サブシステム
- L3 インターコネク
- HPS ペリフェラル
- HPS-to-FPGA ユーザークロック

HPS クロックのプランニングは、ボードレベルのクロック・プランニング、デバイスの FPGA 部分のクロック・プランニング、および HPS ペリフェラルの外部インターフェイスのプランニングに依存します。よって、ボードデザインを確定する前に HPS クロックのコンフィグレーションを検証することが重要です。

**ガイドライン:** プラットフォーム・デザイナーを使用し、MPU およびペリフェラルのクロックを検証します。

プラットフォーム・デザイナーを使用し、HPS コンポーネントのコンフィグレーションを最初に定義します。HPS 入力クロック、ペリフェラル・ソース・クロックおよび周波数を設定します。プラットフォーム・デザイナーの警告またはエラーメッセージに注意し、クロックの設定を変更するか、警告がアプリケーションに悪影響を及ぼさないことを確認することで対応します。

### 2.1.2. 早期ピン・プランニングおよび I/O 割り当ての解析

HPS クロック入力は、イーサネット、大容量ストレージ・フラッシュ、UART コンソールなどの HPS ペリフェラルからの I/O と共有される HPS 専用 I/O バンクにあります。このバンク内での位置はユーザーがコンフィグレーションできます。

**ガイドライン:** HPS 専用 I/O の I/O 電圧レベルを選択します。

HPS 専用 I/O は、1.8 V の電圧レベルをサポートする LVCMOS/LVTTL です。HPS 専用 I/O バンクを使用するようにコンフィグレーションされる HPS ペリフェラル・インターフェイス (イーサネット PHY、UART コンソールなど) および HPS のボードレベルのクロック回路が、1.8 V LVCMOS 信号と互換性があることを確認してください。

### 2.1.3. HPS クロック、リセット、PoR のピン機能と接続

HPS クロックピンとオプションのリセットピンには、ボードレベルのリセットロジックおよび回路のプランニングとデザインを行う際に考慮すべき特定の機能的な動作と要件があります。

**ガイドライン:** HPS クロック入力のピン位置を選択します。

HPS\_OSC\_CLK は、HPS 専用 I/O バンクの任意の位置に配置できます。HPS プラットフォーム・デザイナーのコンポーネントを使用して HPS\_OSC\_CLK のピンを選択し、そのバンクに割り当てられているほかの HPS ペリフェラル I/O の位置との互換性を確認します。

**ガイドライン:** nCONFIG および HPS\_COLD\_nRESET の最小アサート時間仕様に従います。

nCONFIG と HPS\_COLD\_nRESET ピンのリセット信号は、インテル Stratix 10 デバイス・データシートの HPS の章で指定されている最小時間のあいだアサートする必要があります。

**ガイドライン:** HPS\_COLD\_nRESET を SDM QSPI リセットに接続しないでください。

HPS\_COLD\_nRESET は、HPS およびそのペリフェラルのコールドリセット手順を開始するために SDM に入力される双方向ピンです。HPS\_COLD\_nRESET 出力は、HPS がリセットされる際にリセットする必要があるボード上のほかのデバイスをリセットするために使用することができます。ただし、SDM はソフトウェアを介して QSPI のリセットを処理します。HPS\_COLD\_nRESET を SDM QSPI リセットに接続すると、未定義のシステム動作が発生する可能性があります。

### 2.1.4. リモート・システム・アップデート (RSU) 機能に向けた Direct-to-Factory ピンのサポート

インテル Stratix 10 SoC は、リモート・アップデート・システム (RSU) 機能をサポートしています。この機能を使用する場合、複数のプロダクション・イメージをフェイルセーフのファクトリー・イメージとともに外部 SDM フラッシュに保存するオプションが使用できます。POR が終了すると、SDM は特定のシーケンスでプロダクション・イメージのロードを試みます。プロダクション・イメージすべてのロードに失敗した場合、フェイルセーフのファクトリー・イメージがロードされます。



**ガイドライン: Direct-to-Factory イメージピンを使用し、POR 終了時にファクトリー・イメージもしくはアプリケーション・イメージをロードするよう SDM に指示します。**

Direct-to-Factory イメージは、RSU 機能で利用できるオプションのピンです。<sup>(1)</sup>このピンが POR 中にアサートされると、SDM はプロダクション・イメージのロードを試みるのではなく、外部 SDM フラッシュから直接ファクトリー・イメージをロードします。

HPS を RSU 機能とともに使用方法の詳細は、*Intel Stratix 10 SoC Remote System Update (RSU) User Guide* を参照ください。

#### 関連情報

[Intel Stratix 10 SoC Remote System Update \(RSU\) User Guide](#)

### 2.1.5. 内部クロック

HPS クロックのコンフィグレーション・プランニング・ガイドラインの説明に従い HPS クロックのコンフィグレーションを検証後、ソフトウェア制御下で HPS クロックの設定を実装する必要があります。これは通常、ブートローダー・ソフトウェアによって行われます。また、HPS と FPGA 間でリファレンス・クロックを転送するためのガイドラインにも従う必要があります。

**ガイドライン: HPS と FPGA 間で PLL をカスケード接続しないでください。**

FPGA と HPS 間で PLL をカスケード接続することは特性評価されていません。ジッター解析を行わない限り、FPGA と HPS PLL をチェーン接続しないでください。HPS からの出力クロックは、FPGA の PLL に供給されることを目的にしていません。

ソフトウェア制御下で HPS PLL およびクロックを管理するには特定の要件があります。

詳細は、*Intel Stratix 10 Hard Processor System Technical Reference Manual* の「Clock Manager」の章を参照ください。

#### 関連情報

[Intel Stratix 10 Hard Processor System Technical Reference Manual](#)

要求されるソフトウェア・フローの詳細については、「Clock Manager」の章と、特定のペリフェラルおよびサブシステムの章を参照してください。

### 2.1.6. HPS ペリフェラルのリセット管理

HPS ペリフェラルおよびサブシステムには、特定のリセットシーケンス要件があります。SoC EDS で提供されるブートローダー・ソフトウェアは、リセット・マネージャーの章に記載されている要件に基づきリセット管理シーケンスを実装します。

**ガイドライン: SoC EDS の最新のブートローダー・ソフトウェアを使用し HPS リセットを管理します。**

要求されるソフトウェア・フローに関する詳細は、*Intel Stratix 10 Hard Processor System Technical Reference Manual* の「Reset Manager」の章を参照ください。

#### 関連情報

[Intel Stratix 10 Hard Processor System Technical Reference Manual](#)

---

(1) ファクトリー・イメージおよびアプリケーション・イメージはどちらも SDM フラッシュに保存されます。

## 2.2. デバイス I/O を HPS ペリフェラルおよびメモリーに接続するためのデザインにおける考慮事項

HPS をコンフィグレーションする際の最も重要な考慮事項の 1 つは、インテル Stratix 10 SoC デバイス上での I/O 構成を理解することです。

### 1. HPS 専用 I/O

これらの 48 の I/O は物理的に HPS に配置され、HPS に専用のもので、また、HPS クロックおよび、大容量ストレージ・フラッシュ・メモリーを含むペリフェラルに使用されます。

### 2. SDM 専用 I/O

SDM には 24 の専用 I/O があり、それには JTAG、クロック、リセット、コンフィグレーション、リファレンス電圧、ブートおよびコンフィグレーション・フラッシュ・インターフェイス、MSEL が含まれます。

### 3. HPS EMIF I/O

SDRAM メモリーに接続可能なモジュラー I/O バンクは 3 つあります。そのうちの 1 つは、アドレス、コマンド、および ECC データ信号の接続に使用されます。残りの 2 つのバンクは、データ信号を接続するためのものです。

### 4. FPGA I/O

汎用 I/O は FPGA ロジック、FPGA 外部メモリー・インターフェイス、および高速シリアル・インターフェイスに使用することができます。ほとんどの HPS ペリフェラル・インターフェイスは、FPGA ファブリックにエクスポートし、FPGA I/O へのカスタム適応およびルーティングが可能です。

以下の表は、各 I/O タイプの特性をまとめています。





表 2. SoC-FPGA I/O タイプの要約

	HPS 専用 I/O	SDM 専用 I/O	HPS EMIF I/O	FPGA I/O
利用可能な I/O 数	48	24	3 つの I/O 48 バンク	ほかのすべてのデバイス I/O
位置	HPS 内	SDM 内	FPGA I/O バンク 2L、2M、2N	I/O 列は FPGA デバイス内にあります
サポートされる電圧	1.8 V	1.8 V	DDR3 および DDR4 プロトコルの LVDS I/O バンクサポート	LVDS I/O バンク、3V I/O バンクおよび高速シリアル・トランシーバー
用途	HPS クロック、HPS ペリフェラル、大容量ストレージ・フラッシュ、HPS JTAG	FPGA JTAG (SDM 専用ピンを使用)、クロック、リセット、コンフィグレーション、リファレンス電圧、ブートおよびコンフィグレーション・フラッシュ・インターフェイス	HPS メインメモリー	汎用およびトランシーバー I/O
タイミング制約	固定	固定	メモリー・コントローラー IP による	ユーザー定義
推奨されるペリフェラル	HPS ペリフェラル I/O (イーサネット PHY、USB PHY、NAND や SD/MMC などの大容量ストレージ・フラッシュ、TRACE デバッグなど)	ブートおよびコンフィグレーション・ソース、SDM 専用ピンを介する FPGA JTAG、および MSEL 信号が SDM に接続されます	DDR3、DDR4、SDRAM	低速 HPS ペリフェラル (I <sup>2</sup> C、SPI、EMAC-MII)、FPGA EMIF などの FPGA I/O、高速 LVDS I/O、トランシーバー I/O、その他の並列およびコントロール/ステータス I/O

ブートおよびコンフィグレーション中のコンソール出力に関する詳細は、「UART インターフェイスのデザイン・ガイドライン」の章を参照ください。

#### 関連情報

UART インターフェイスのデザイン・ガイドライン (23 ページ)

### 2.2.1. HPS ピンの多重化におけるデザインの考慮事項

HPS 専用 I/O ピンは合計 48 あります。プラットフォーム・デザイナーの HPS コンポーネントは、ピンの多重化設定と、ほとんどのペリフェラルを FPGA ファブリックにルーティングするオプションを提供します。

**ガイドライン:** USB を先頭に、USB、EMAC およびフラッシュ・インターフェイスを HPS 専用 I/O に最初にルーティングします。

インテルではまず、USB、イーサネット、フラッシュなどの高速インターフェイスを専用 I/O にルーティングすることから始めることを推奨しています。

### 2.2.2. HPS I/O の設定: 制約とドライブ強度

**ガイドライン:** HPS 専用 I/O に向けて I/O 設定を正しくコンフィグレーションしていることを確認してください。

HPS ピン位置の割り当ては、HPS を含むプラットフォーム・デザイナー・システムを生成する際に自動的に管理されます。同様に、HPS SDRAM インターフェイスのタイミングおよび I/O 制約は、HPS IP に向けた インテル Stratix 10 外部メモリー・インターフェイスによって管理されます。HPS 専用 I/O に対す

る I/O 制約 (ドライブ強度、ウィーク・プルアップ・イネーブル、終端設定) を、インテル Quartus Prime 開発ソフトウェアを使用し FPGA I/O と同じ方法で管理する必要があります。FPGA I/O を使用するようにコンフィグレーションされるペリフェラルもまた、ピン位置を含め インテル Quartus Prime 開発ソフトウェアで完全に制約される必要があります。

## 2.3. HPS インターフェイスにおけるデザイン・ガイドライン

この章では、EMAC、USB、SD/MMC、NAND、UART、および I<sup>2</sup>C などの HPS インターフェイスに対するデザイン・ガイドラインについて説明します。

### 2.3.1. HPS EMAC PHY インターフェイス

Synopsys\* DesignWare\* 3504-0 Universal 10/100/1000 Ethernet MAC IP バージョンに基づく EMAC は 3 つあります。プラットフォーム・デザイナーで HPS コンポーネントを EMAC ペリフェラルにコンフィグレーションする際は、HPS 専用 I/O バンク<sup>(2)</sup>にあるサポートされている次の PHY インターフェイスから 1 つを、各 EMAC インスタンスに選択する必要があります。

- Reduced Media Independent Interface (RMII)
- Reduced Gigabit Media Independent Interface (RGMII)

#### 注意:

RGMII- 内部遅延 (RGMII-ID) は RGMII バージョン 2.0 仕様のコンポーネントで、転送されたデータバスクロックに遅延を追加し、そのクロックをデータの中央に配置するトランスミッターの機能を定義します。遅延は、トランスミッター (MAC) の TX\_CLK およびトランスミッター (PHY) の RX\_CLK に追加されます。HPS 専用 I/O バンクのピン MUX は、150 ps の増分で最大 2.4 ns の遅延を追加する機能を備えています。これは、PHY でクロックを送信データの中央に配置するために必要な 1.5 ns を超えています。

**ガイドライン: PHY デバイスを選択する際は、要求するイーサネット・レート、使用可能な I/O およびトランシーバー、スキュー制御機能を提供する PHY デバイス、およびデバイスドライバーの可用性を考慮してください。**

インテル Stratix 10 SoC 開発キットは Microchip KSZ9031 Ethernet PHY を使用しています。このデバイスは、インテル Stratix 10 HPS イーサネット PHY インターフェイスおよびソフトウェア・デバイス・ドライバーで動作することが分かっています。

HPS コンポーネントによって FPGA ファブリックに公開される MII または GMII PHY インターフェイスは、FPGA のソフト適応ロジックと汎用 FPGA I/O およびトランシーバー FPGA I/O の機能を使用し、RMII、SGMII、SMII、TBI などのほかの PHY インターフェイス標準に適合させることが可能です。

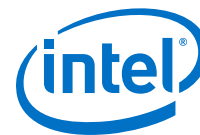
詳細については、ご利用のオペレーティング・システムで使用可能なデバイスドライバー、または インテル Stratix 10 SoC 開発キットで提供される Linux デバイスドライバーを参照してください。

EMAC は、HPS および FPGA I/O を介してさまざまな PHY インターフェイスと制御オプションを提供します。

HPS I/O にピン制限のあるデザインの場合、EMAC は GMII または MII PHY インターフェイスを FPGA ファブリックに公開するようコンフィグレーションできます。これは FPGA I/O ピンに直接ルーティング可能です。PHY インターフェイスを FPGA ファブリックに公開すると、適切な汎用 I/O またはトランシーバー I/O リソースとともにソフトロジックを使用し、GMII または MII を SGMII や RMII などのほかの PHY インターフェイス・タイプに適合させることができます。

---

<sup>(2)</sup> HPS 専用 I/O バンクは、1.8 V 信号の 48 の I/O で構成されます。



**注意:** ギガビットおよび 10/100 Mbps アクセスには、GMII および MII バス・インターフェイスをそれぞれ使用し、FPGA ファブリックを介して PHY を HPS EMAC に接続できます。

**ガイドライン:** GMII-to-SGMII アダプターは、トランシーバー・ベースの SGMII 光モジュールに自動的に適合させるために使用可能です。

EMAC はまた、MDIO の代わりに I<sup>2</sup>C を制御インターフェイスに提供します。HPS もしくは FPGA は、5 つの汎用 I<sup>2</sup>C ペリフェラルのうちの 3 つを PHY デバイスの制御に使用できます。

- i2c\_emac\_0
- i2c\_emac\_1
- i2c\_emac\_2

#### 関連情報

Golden System Reference Design

### 2.3.1.1. PHY インターフェイス

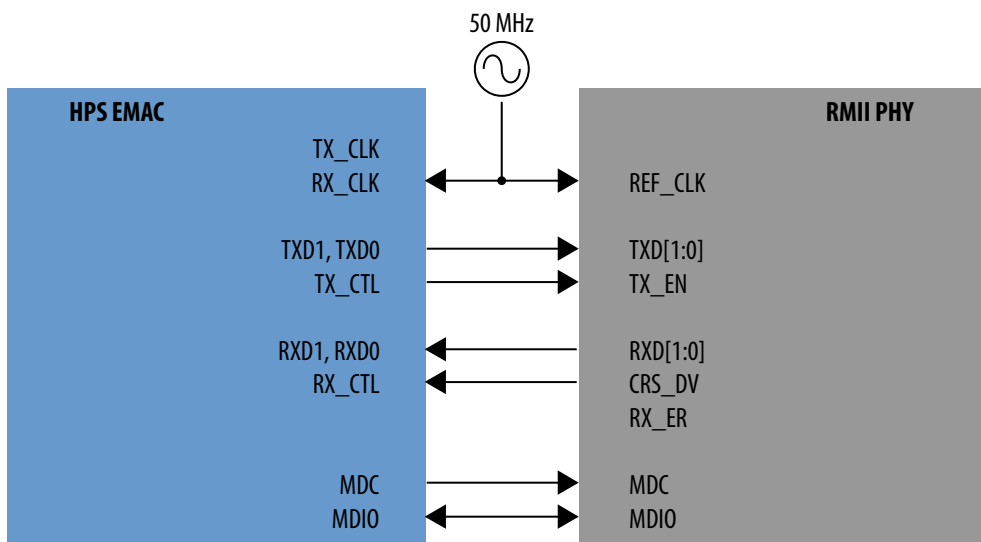
#### RMII および RGMII PHY インターフェイスにおける考慮事項

##### 2.3.1.1.1. RMII

RMII は、システムに同期する 50 MHz の単一中央クロックソース (REF\_CLK) をすべてのポートの送信バスおよび受信バスに使用します。これは、各ポートの TX\_CLK と RX\_CLK のソース・シンクロナスのクロック・ペアではなく、単一のボード・オシレーターをデザインで使用できるため、ポート密度が高いシステムにおいてシステムのクロックを簡素化しピン数を低減します。

RMII は 2 ビット幅の送信および受信データバスを使用します。すべてのデータおよび制御信号は、REF\_CLK 立ち上がりエッジに同期しています。RX\_ER 制御信号は使用されません。10 Mbps モードでは、データおよび制御信号はすべて、10 REF\_CLK クロックサイクル間有効に保たれます。

図 -1: RMII MAC/PHY インターフェイス



### インターフェイス・クロック・スキーム

EMAC および RMII PHY は、50 MHz の REF\_CLK ソースを提供できます。HPS\_OSC\_CLK 入力などの既存のクロックソースを使用することで、内部 PLL はシステムのクロック・デザインをさらに簡素化し、クロックソースの追加を不要にします。

この章では、HPS EMAC または PHY をソースとする REF\_CLK のデザインシナリオを説明します。

**ガイドライン:** アプリケーションの REF\_CLK ソース選択に関する詳細は、PHY データシートを確認してください。

**注意:** 選択した PHY がアプリケーションの REF\_CLK クロック・スキームをサポートしていることを確認してください。PHY のデータシートで指定されている要件および考慮事項に注意してください。

REF\_CLK のソースには、次の 2 つの方法のいずれかを使用することができます。

- HPS をソースとする REF\_CLK
- PHY をソースとする REF\_CLK

#### 図 -2: HPS をソースとする REF\_CLK

このスキームでは、EMAC の HPS RMII I/O TX\_CLK 出力を、HPS RMII I/O RX\_CLK 入力および PHY REF\_CLK 入力に接続します。

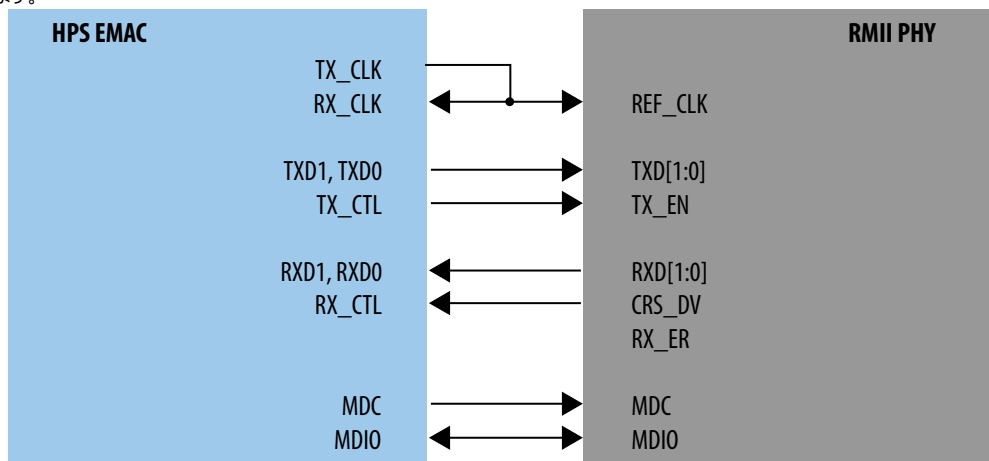
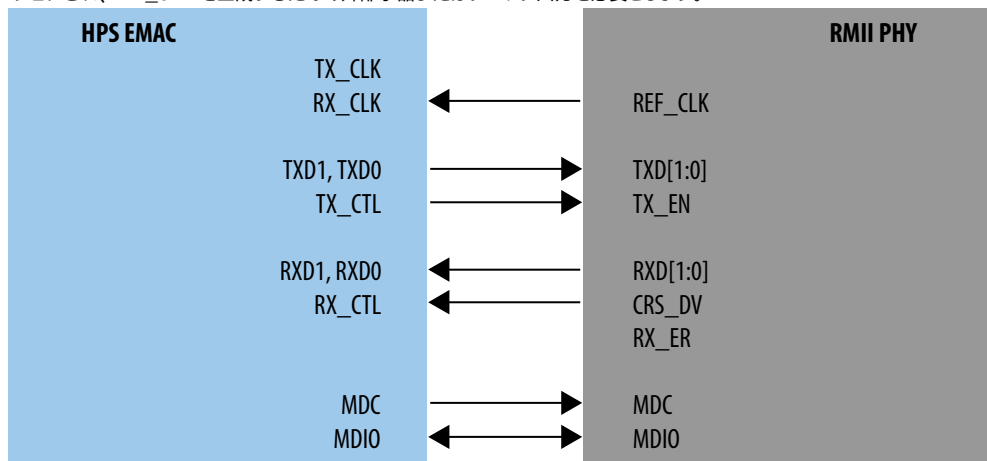


図 -3: PHY をソースとする REF\_CLK

このスキームでは、PHY の REF\_CLK 出力を EMAC の HPS RMII I/O RX\_CLK 入力に接続します。EMAC の HPS RMII I/O TX\_CLK 出力は未接続にします。REF\_CLK ソースを提供できる PHY は通常、ピン・ブートストラップを介してそのようにコンフィグレーションされ、REF\_CLK を生成するための外部水晶またはクロック入力が必要とします。



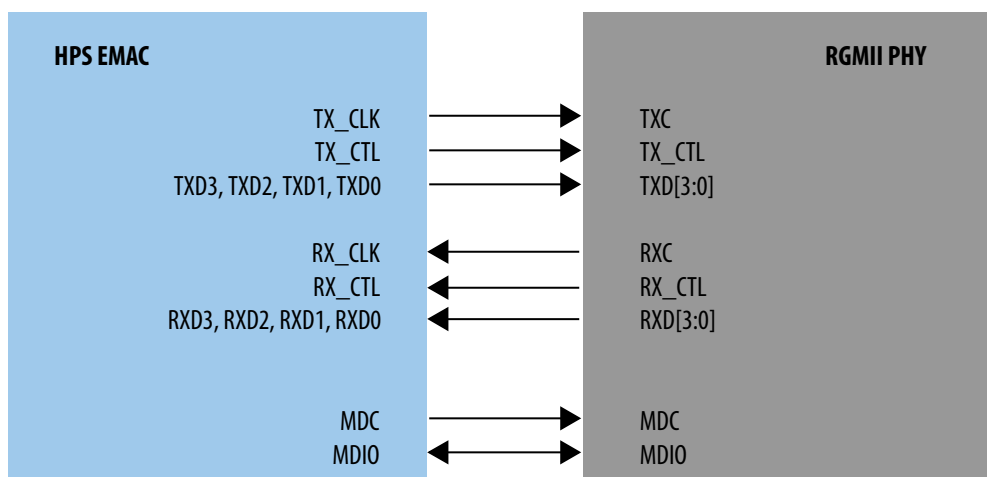
### 2.3.1.1.2. RGMII

RGMII は、PHY 層で 10 Mbps、100 Mbps、および 1000 Mbps の接続速度をサポートするため、最も一般的なインターフェイスです。

RGMII は、4 ビット幅の送信データパスおよび受信データパスを使用しており、それぞれが独自のソース・シンクロナスのクロックを備えています。送信データおよび制御信号はすべて TX\_CLK にソース・シンクロナスで、受信データおよび制御信号はすべて RX\_CLK にソース・シンクロナスです。

すべての速度モードにおいて、TX\_CLK は MAC によってソースされ、RX\_CLK は PHY によってソースされます。1000 Mbps モードでは、TX\_CLK および RX\_CLK は 125 MHz であり、デュアル・データ・レート (DDR) 信号が使用されます。10 Mbps および 100 Mbps モードでは、TX\_CLK および RX\_CLK はそれぞれ 2.5 MHz および 25 MHz であり、立ち上がりエッジのシングル・データ・レート (SDR) 信号が使用されます。

図 -4: RGMII MAC/PHY インターフェイス



## I/O ピンのタイミング

この章では、1000 Mbps モードの要件を満たすという観点から RGMII インターフェイスのタイミングについて説明します。インターフェイスのタイミングマージンは 1000 Mbps モードにおいて最も要求が厳しいため、これはここで考慮する唯一のシナリオです。

125 MHz における周期は 8 ns ですが、両方のエッジが使用されるため、有効周期はわずか 4 ns になります。TX バスおよび RX バスは独立しておりソース・シンクロナスであるため、タイミングが簡潔になります。RGMII の仕様では、レシーバー側で CLK が DATA から一方の方向に最小 1.0 ns および最大 2.6 ns 遅延するように要求しています。

すなわち、TX\_CLK は MAC 出力から PHY 入力に、RX\_CLK は PHY 出力から MAC 入力に遅延する必要があります。信号は、出力ピンで測定された場合に各方向 +/- 500 ps RGMII スキュー仕様内でソース・シンクロナスに送信されます。それぞれの方向に必要な最小遅延は 1 ns ですが、インテルでは 1.5 ns から 2.0 ns の遅延をターゲットとし、大幅なタイミングマージンを確保することを推奨しています。

### 送信バスのセットアップおよびホールド

送信に重要なのは、TX\_CTL と TXD[3:0] に対する TX\_CLK のセットアップおよびホールドのみです。インテル Stratix 10 の I/O は、出力で最大 2.4 ns の追加遅延を 150 ps の増分で提供することができます。この遅延は、インテル Quartus Prime の Assignment Editor の出力遅延ロジックオプションを使用し有効にします。

**ガイドライン: Stratix 10 からの TX\_CLK の場合、1.8 ns の I/O 遅延を導入し、RGMII 仕様にある 1.0 ns の PHY 最小入力セットアップおよびホールド時間を満たす必要があります。**

インテル Stratix 10 SoC HPS 専用 I/O および FPGA I/O は、最大 2.4 ns の追加出力遅延を 150 ps の増分でサポートします。HPS 専用 I/O を使用する場合に MAC の TX\_CLK 出力に追加される遅延は、HPS プラットフォーム・デザイナーの IP コンポーネントでコンフィグレーションできます。

**ガイドライン: 要求される遅延を HPS EMAC 出力にコンフィグレーションするために必要な Intel 設定がデザインに含まれていることを確認してください。**

インテル Stratix 10 SoC 開発キットおよび関連する インテル Stratix 10 ゴールデン・ハードウェア・リファレンス・デザイン (GHRD は GSRD のハードウェア・コンポーネントです) において、TX\_CLK の出力遅延設定の設定を行う例は、HPS プラットフォーム・デザイナーの IP コンポーネント・コンフィグレーションにあります。

### 受信バスのセットアップおよびホールド

受信タイミングを考慮する際に必要なのは、RX\_CTL と RXD[3:0] に対する RX\_CLK のセットアップおよびホールドのみです。インテル Stratix 10 の I/O は、最大 3200 ps の遅延を入力に追加できます。インテル Stratix 10 の入力の場合、3.2 ns の I/O 遅延は、PHY 側もしくはボードトレースの遅延側のほかの内容を考慮することなく RX\_CLK のこのタイミングを達成できます。

**ガイドライン: PHY が RGMII-ID をサポートしていない場合、Stratix 10 SoC HPS 専用 I/O もしくは FPGA I/O のコンフィグレーション可能な遅延要素を使用し、RX\_CLK を RX\_DATA および CTL データの有効ウィンドウの中央に配置します。**

HPS I/O を使用する場合、HPS プラットフォーム・デザイナーの IP コンポーネントで RX\_CLK の遅延をコンフィグレーションします。FPGA I/O を使用する場合は、プロジェクト設定ファイル (.qsf) の入力遅延設定で RX\_CLK 入力に遅延を追加します。



### 2.3.1.2. FPGA I/O を介して接続される PHY インターフェイス

HPS EMAC PHY インターフェイスに FPGA I/O を使用することは、PHY インターフェイスに対応する空きが十分でない場合や、HPS EMAC でネイティブにサポートされていない PHY インターフェイスに適用する場合に有効です。

**ガイドライン: プラットフォーム・デザイナーで HPS コンポーネントをコンフィグレーションする際は、PHY インターフェイスの送信クロック周波数を指定します。**

他の PHY インターフェイスに適応させる場合も含め、GMII もしくは MII には、HPS EMAC PHY インターフェイスの最大送信バスクロック周波数を指定します。この設定により、プラットフォーム・デザイナーのシステム生成時に、適切なクロックタイミング制約が PHY インターフェイスの送信クロックに適用されます。

#### 2.3.1.2.1. GMII および MII

GMII および MII は、EMAC 信号を FPGA コア・ルーティング・ロジックに駆動し、最終的に FPGA I/O ピンまたは FPGA コアの内部レジスターに駆動することで、インテル Stratix 10 でのみ使用可能です。

**ガイドライン: タイミング制約を適用し、Timing Analyzer でタイミングを検証します。**

配線遅延は FPGA コアと I/O 構造で大きく異なる可能性があるため、タイミングレポートを確認し、特に GMII の場合は、タイミング制約を作成することが重要です。GMII は 125 MHz のクロックを備えており、RGMII とは異なり単一データレートです。ただし GMII では、CLK と DATA のスキューに関して RGMII と同じ内容を考慮する必要はありません。信号はネガティブエッジで起動され、立ち上がりエッジでキャプチャーされることで、デザインによって自動的に中央に配置されます。

**ガイドライン: インターフェイス I/O は FPGA I/O 境界で登録します。**

コアおよび I/O の遅延は 8 ns を簡単に超えるため、インテル ではこれらのバスを各方向で I/O エレメント (IOE) レジスターに登録し、それらがコア FPGA ロジック・ファブリックを移動する際にアライメントが維持されるようにすることを推奨しています。送信データと制御では、HPS EMAC からの `emac[0,1,2]_gtx_clk` 出力の立ち下がりエッジでこれらの信号をラッチすることにより、`clock-to-data/control` の関係を維持します。受信データと制御は、FPGA I/O 入力において PHY をソースとする `RX_CLK` の立ち上がりエッジでラッチします。

**ガイドライン: MII モードの送信タイミングを検討します。**

MI 2 は、PHY が 100 Mbps モードの場合は 25 MHz、PHY が 10 Mbps モードの場合は 2.5 MHz であるため、最短クロック周期は 40 ns です。PHY は、送信方向と受信方向の両方にクロックを供給します。送信タイミングは PHY によって供給される `TX_CLK` クロックに相対するため、ターンアラウンド・タイムが懸念されることがありますが、40 ns の長いクロック周期のため通常問題ではありません。

リファレンス・クロックは FPGA を介して送信され、その後データに出力されます。15 ns の入力セットアップ時間を考慮すると、往復の遅延は 25 ns 未満でなければなりません。送信データと制御は、PHY をソースとする `TX_CLK` のネガティブエッジで HPS EMAC 送信パスロジックによって FPGA ファブリックに起動されます。これにより、40 ns のクロックからセットアップまで (Clock-to-Setup) のタイミング・バジェットのうち 20 ns が取られることに注意してください。

PHY から SoC ボードの伝播遅延を引き起こすデータ到着タイミングにおける往復のクロック・パス遅延に加え、SoC ピンから HPS EMAC 送信クロック・マルチプレクサーでの内部パス遅延が残りの 20 ns のセットアップ・タイミング・バジェットを消費するため、MII モードの送信データと制御では、場合によっては FPGA ファブリックの `phy_txclk_o` クロック出力レジスターの立ち上がりエッジに、送信データと制御をリタイミングする必要があります。

#### 2.3.1.2.2. RGMII への適合

インテル Stratix 10 SoC デバイスは、FPGA I/O ピンを使用する HPS EMAC 信号の RGMII への適合をサポートしていません。

#### 2.3.1.2.3. RMII への適合

FPGA のロジックを使用し、MII HPS EMAC PHY 信号を FPGA I/O ピンで RMII PHY インターフェイスに適合させることができます。

**ガイドライン: 50 MHz の `REF_CLK` ソースを提供します。**

RMII PHY は、単一の 50 MHz リファレンス・クロック (`REF_CLK`) を送信と受信のデータおよび制御に使用します。ボードレベルのクロックソース、FPGA ファブリックから生成されるクロック、もしくは `REF_CLK` の生成が可能な PHY のいずれかで 50 MHz の `REF_CLK` を提供します。

**ガイドライン: 送信と受信のデータおよび制御パスを適応させます。**

FPGA ファブリックで公開される HPS EMAC PHY インターフェイスは MII であり、2.5 MHz および 25 MHz の個別の送信クロック入力と受信クロック入力を、それぞれ 10 Mbps および 100 Mbps の動作モードで必要とします。送信データパスと受信データパスはどちらも 4 ビット幅です。RMII PHY は 10 Mbps と 100 Mbps の両方の動作モードにおいて、50 MHz の `REF_CLK` を送信データパスと受信データパスに使用します。RMII の送信および受信データパスは 2 ビット幅です。10 Mbps において、送信および受信のデータと制御は 10 クロックサイクルの 50 MHz `REF_CLK` 間安定します。FPGA ファブリックの適応ロジックを提供し、HPS EMAC MII と外部 RMII PHY インターフェイスを適応させる必要があります。すなわち、25MHz および 2.5 MHz の 4 ビットと 50 MHz の 2 ビットを適応させ、10 Mbps モードにおいては 10 倍のオーバーサンプリングを行います。

**ガイドライン: HPS EMAC MII の `tx_clk_in` クロック入力でグリッチのないクロックソースを提供します。**

HPS コンポーネントの MII インターフェイスは、`emac[0,1,2]_tx_clk_in` 入力ポートで 2.5 および 25 MHz の送信クロックを必要とします。2.5 MHz と 25 MHz の切り替えは、HPS EMAC の要求に応じてグリッチなしで行う必要があります。FPGA PLL を使用し 2.5 MHz および 25 MHz の送信クロックを提供し、それに加えて `ALTCLKCTRL` IP ブロックでカウンター出力をグリッチなしで選択することが可能です。

#### 2.3.1.2.4. SGMII への適合

GMII-to-SGMII アダプターコアを使用し、FPGA トランシーバー I/O ピンで GMII HPS EMAC PHY 信号を Serial Gigabit Media Independent Interface (SGMII) PHY インターフェイスに適合させることができます。これには FPGA のロジックとマルチギガビット・トランシーバー I/O またはソフト CDR モードの LVDS SERDES を使用します。この適合にカスタムロジックをデザインすることは可能ですが、この章ではプラットフォーム・デザイナーのアダプター IP を使用する方法を説明します。





**ガイドライン:** プラットフォーム・デザイナーで利用可能な **GMII-to-SGMII アダプター IP** を使用します。

プラットフォーム・デザイナーで HPS コンポーネントを EMAC 「To FPGA」 I/O インスタンスにコンフィグレーションし、GMII を PHY インターフェイス・タイプとして管理インターフェイスとともに選択します。生成される HPS コンポーネントの GMII 信号をプラットフォーム・デザイナーでエクスポートしないでください。代わりに、インテル GMII-to-SGMII アダプター IP をプラットフォーム・デザイナーのサブシステムに追加し、HPS コンポーネントの GMII 信号に接続します。GMII-to-SGMII アダプター IP は、プラットフォーム・デザイナーのインテル HPS EMAC インターフェイス・スプリッター IP を使用し、「emac」コンジットを HPS コンポーネントから切り離し、GMII-to-SGMII アダプターで使います。アダプター IP は、1000BASE-X/SGMII PCS PHY-only モード（つまり、ソフト MAC コンポーネントなし）でコンフィグレーションされたインテル Triple Speed Ethernet (TSE) MAC IP をインスタンス化します。インテル GMII-to-SGMII アダプター IP の使用方法に関する詳細は、*Embedded Peripherals User Guide* を参照ください。

**ガイドライン:** **1000BASE-X PCS オプションでの TSE MAC IP はトランシーバー I/O のオプションを提供しなくなったため、FPGA トランシーバー I/O を使用して SGMII PHY インターフェイスをインテル Stratix 10 HPS EMAC インスタンスに実装するには、「NONE」を PCS I/O オプションに選択します。これにより TBI インターフェイスが提供されます。トランシーバー PHY IP は、インテル Stratix 10 で個別にインスタンス化および接続する必要があります。**

#### 関連情報

[Embedded Peripherals IP User Guide](#)

### 2.3.1.3. MDIO

インテル MDIO (Management Data Input/Output) PHY 管理バスには、各 MAC ごとに MDC と MDIO の 2 つの信号があります。MDC はクロック出力で、フリーランニングではありません。2.5 MHz での最小周期は 400 ns です。MDIO は、High-Z バスのターンアラウンド周期を備えた双方向データ信号です。

MAC が PHY に書き込むと、データは立ち下がりエッジで起動されます。つまり、200 ns -10 ns = 190 ns がフライトタイム、信号の整定、およびレシーバーでのセットアップにあります。また、データは次のネガティブエッジまで切り替わらないため、200 ns のホールド時間があります。これらの要件は、ほぼすべてのボードトポロジーで簡単に満たすことができます。MAC が PHY から読み取る場合、PHY は読み取りデータを 0 から 300 ns で MAC に戻し、100 ns から 10 ns を引いたセットアップ時間、つまり 90 ns をフライトタイム、信号の整定、およびレシーバーでのセットアップに残す必要があります。この要件もまた、非常に簡単に満たすことができます。

**ガイドライン:** **MDC および MDIO のボードプルアップ**

どちらの信号も外部プルアップ抵抗を必要とします。正確なプルアップ抵抗値については、お使いの PHY のデータシートを確認してください。1K  $\Omega$  が一般的な抵抗値です。

**ガイドライン:** **MDIO が要求するインターフェイスのタイミングを確認します。**

MDIO は、MDC に関して 10 ns のデータのセットアップとホールド時間を必要とします。

### 2.3.1.4. PHY インターフェイス・デザインの一般的な考慮事項

#### 2.3.1.4.1. シグナル・インテグリティ

**ガイドライン: SoC デバイスのオンチップ終端 (OCT) を利用します。**

インテル Stratix 10 デバイスは、多くの設定に合わせて出力を調整できます。50  $\Omega$  の出力インピーダンスが多くの場合において最適な値です。インテル Quartus Prime は、RGMIIC 出力でのキャリブレーションなしに自動的に直列 OCT を使用します。インテル Quartus Prime のフィッターレポートを確認し、インターフェイスの出力の OCT 設定を検証してください。

**ガイドライン: 適切なボードレベルの終端を PHY 出力で使用します。**

いくつかの PHY のみが出力の I/O 調整を提供するため、インテル では、シミュレーターを使用し、インテル Stratix 10 デバイスへの信号パスを検証することを推奨しています。必要に応じて PHY 出力ピン付近の各信号に直列抵抗を配置し、反射を減らします。

**ガイドライン: PHY TX\_CLK および EMAC RX\_CLK 入力での反射を最小限に抑え、ダブルクロッキングを防ぎます。**

接続が「T」としてルーティングされる場合、REF\_CLK ロードでダブルエッジが見られないようにシグナル・インテグリティを維持する必要があることに注意してください。REF\_CLK ロードでの反射を最小限に抑え、ダブルクロッキングを防ぎます。

**ガイドライン: シグナル・インテグリティ (SI) シミュレーション・ツールを使用します。**

SI シミュレーションは、これらの単方向信号で容易に使用できます。これらの信号はほとんどの場合においてポイントツーポイントであるため、各信号に配置する適切な直列抵抗を決定するだけで通常は十分です。多くの場合この抵抗は必要ありませんが、これを決定する際はデバイスのドライブ強度とトレース長、およびトポロジーを検討する必要があります。

### 2.3.2. USB インターフェイスのデザイン・ガイドライン

インテル Stratix 10 HPS では、1.8 V の専用 HPS I/O を使用し、組み込み USB MAC を業界標準の USB 2.0 ULPI PHY に直接接続することができます。FPGA 配線リソースを使用せず、タイミングが固定されているため、デザインは簡潔になります。

このガイドでは、サポートされているすべての PHY 動作速度 (高速 HS 480 Mbps、フルスピード FS 12 Mbps、および低速 LS 1.5 Mbps) を網羅するデザイン・ガイドラインについて説明します。

**ガイドライン: インテル では、デバイスがクロックを供給する場合の USB PHY モードと、外部クロックがソースの場合の USB PHY モードをどちらもサポートするボードをデザインすることを推奨しています。**

Stratix 10 SoC 開発キットは Microchip USB3320 USB PHY を使用します。このデバイスは、HPS USB モジュールで動作することが知られています。

インテル Stratix 10 SoC の ULPI MAC と PHY 間のインターフェイスは、8 ビットのデータと次の制御信号で構成されています。

- STP
- NXT
- DIR

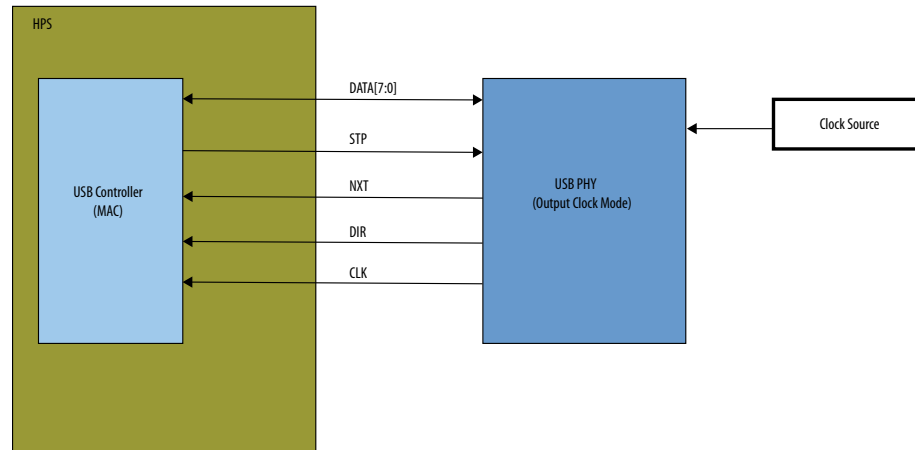


最後に、60 MHz の静的クロックが PHY または外部オシレーターから駆動され、一部の HPS から USB MAC のレジスターアクセスなどの動作に必要とされます。PHY メーカーより提供されているリセットおよび電源投入に関する推奨事項に従っていることを確認してください。

USB PHY が入力クロックモードと出力クロックモードをどちらもサポートする場合、インテルでは、ボードをどちらのモードもサポートするようにデザインし、潜在的なタイミング問題を軽減できるようにすることを推奨します。通常これらのモードは、High または Low にプルされる受動ブートストラップ・ピンで選択されます。

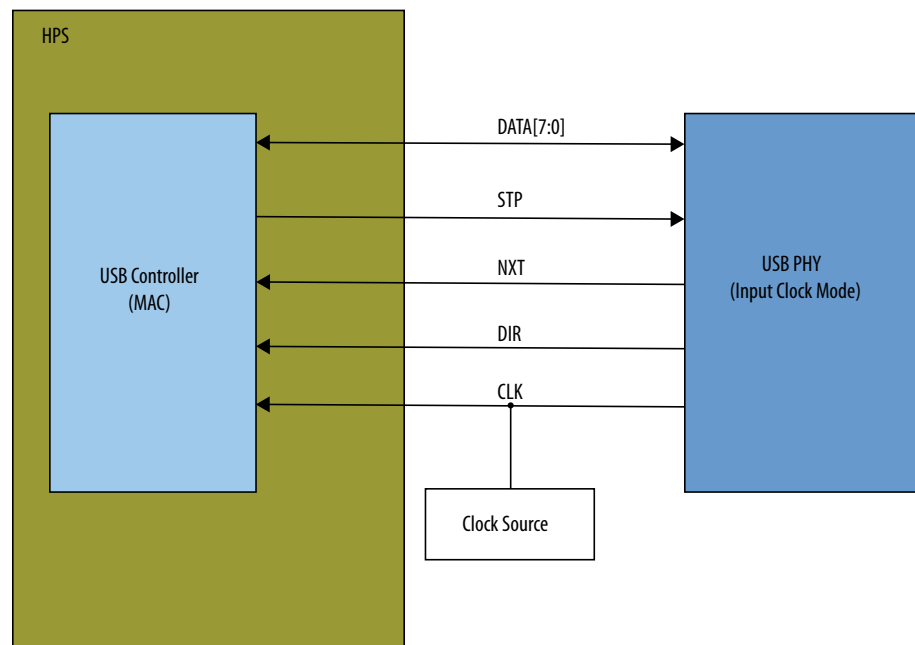
- 出力モード—出力クロックモードでは、クロックは USB PHY によって生成されます。信号はすべてこのクロックに同期しています。

図 -5: 出力モード



- 入力モード—入力クロックモードでは、PHY は外部ソースからのクロックを受信します。信号はすべてこのクロックに同期しています。このモードにおいてクロックは、FPGA の PLL または外部ソースで生成できます。

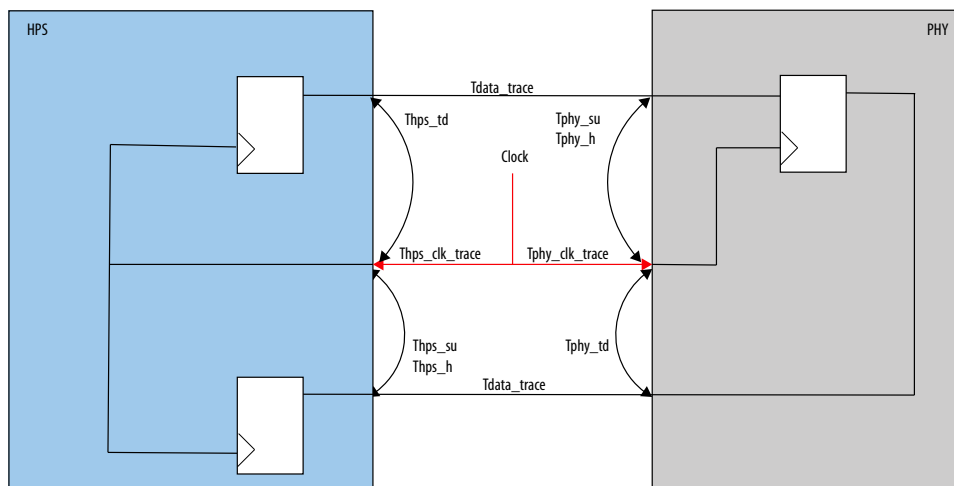
図 -6: 入力モード



**ガイドライン: USB 信号のトレース長が最小になっていることを確認します。**

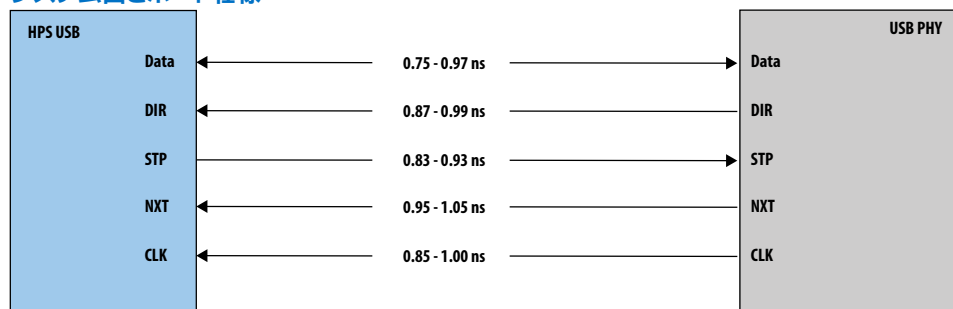
60 MHz における周期は 16.67 ns であり、その間に、例えばクロックは外部 PHY から MAC に移動し、そしてデータおよび制御信号は MAC から PHY に移動する必要があります。往復遅延があるため、クロックおよび ULPI 信号の最大長は重要です。予備のタイミングデータに基づき、最大長は 7 インチ未満にすることが推奨されます。これは 5 ns の  $T_{CO}$  仕様の PHY に基づいています。仕様がそれよりも遅い場合は、それに応じて全長を短くする必要があります。

図 -7: トレース長



バスの USB PHY 側にセットアップ・タイミングのマーヅンがほとんどない場合、PHY を入力クロックモードに切り替え、60 MHz クロックソースをボードから供給することができる場合があります。

図 -8: システム図とボード仕様



**ガイドライン: シグナル・インテグリティが考慮されていることを確認します。**

シグナル・インテグリティは主に、PHY から HPS の MAC に駆動される CLK 信号において重要です。これらの信号は最大長のポイントツーポイントであるため、通常は終端せずに実行できますが、インテルでは、トレースをシミュレーションし、反射を最小限に抑えることを推奨しています。シミュレーションで特に示されない限り、FPGA からの 50  $\Omega$  出力設定を使用することが一般的に推奨されます。可能であれば、PHY ベンダーが提供する同様の設定を使用します。

**ガイドライン: OTG 動作を使用する場合はデザインを適切に行います。**

On-the-Go (OTG) 機能を使用する場合、SoC はホストまたはエンドポイントになることができます。ホストモードにおいては、USB フラッシュドライブをサポートしている場合や、潜在的に USB ハードドライブをサポートしている場合などの電力供給を考慮してください。これらの電力要件と逆電流は通常、インテル Stratix 10 SoC FPGA 開発キットで使用されているような外部ダイオードと電流リミッターを使用し考慮する必要があります。

インテル Stratix 10 SoC 開発ボードの回路図に関する内容は、*Stratix 10 FPGA Development Kit User Guide* を参照ください。

#### 関連情報

[Stratix 10 SoC Development Board Schematics](#)

### 2.3.3. SD/MMC および eMMC カード・インターフェイスのデザイン・ガイドライン

ハード・プロセッサ・システム (HPS) に接続される Synopsys DesignWare に基づくセキュア・デジタル・マルチメディア・カード (SD/MMC) コントローラーは、大容量ストレージに使用されます。このモジュールは以下をサポートします。

- SD バージョン 3.0 に加え 3.01
- Embedded MMC (eMMC) バージョン 4.5<sup>(3)</sup> に加え 4.51 および 5.0

**ガイドライン: 1.8V の SD カードを使用している場合、電圧変換トランシーバーが適切に実装されていることを確認してください。**

HPS I/O は、1.8 V の固定電圧レベルを使用します。多くの SD カードには、1.8 V または 3.3 V で信号を送るオプションがありますが、初期電源投入時の電圧要件は 3.3 V です。3.3V の SD カードを使用する場合は、電圧切り替えが必要です。カードに電力を供給するための正しい電圧レベルを得るには、電圧変換トランシーバーが必要です。

*Stratix 10 Hard Processor System Technical Reference Manual* の「SD/MMC Controller」の章にある Voltage Switching の項目で説明されているガイドラインに従ってください。

**表 3. レベルシフターの要件**

HPS I/O バンクの電圧	SD カードの電圧	レベルシフターの必要性
1.8 V	3.0 V	あり
1.8 V	1.8 V	なし

#### 関連情報

[SD/MMC Controller](#)

「Voltage Switching」の章に記載されているガイドラインに従ってください。

### 2.3.4. フラッシュ・インターフェイスのデザイン・ガイドライン

**ガイドライン: QSPI フラッシュを SoC デバイスに接続します。**

HPS には QSPI フラッシュ・コントローラーがありません。HPS は、SDM の QSPI コントローラーにアクセスできます。

<sup>(3)</sup> HS400 モードはサポートされていません。



インテル Stratix 10 SoC 開発キットは、MT25QU02GCBB8E12-0SIT QSPI フラッシュメモリーを使用しています。このデバイスは、SDM QSPI コントローラーで動作することが分かっています。

QSPI フラッシュを SDM QSPI インターフェイスに接続する際の考慮事項については、*Intel Stratix 10 Hard Processor System Technical Reference Manual* を参照ください。

#### 関連情報

[Intel Stratix 10 Hard Processor System Technical Reference Manual](#)

### 2.3.4.1. NAND フラッシュ・インターフェイスのデザイン・ガイドライン

**ガイドライン:** 選択した NAND フラッシュデバイスが、8 ビットまたは 16 ビットの ONFI 1.0 に準拠するデバイスであることを確認してください。

HPS の NAND フラッシュ・コントローラーには以下が必要です。

- 8 ビットまたは 16 ビットの ONFI 1.0 に準拠する外部フラッシュデバイス
- 大容量ストレージ使用に向けた x16 のサポート
- シングルレベル・セル (SLC) もしくはマルチレベル・セル (MLC)
- ce# と rb# ピンのペアは 1 組のみブートソースに利用可能です。大容量ストレージには 3 組まで追加ペアを使用できます。
- ページサイズ: 512 バイト、2 KB、4 KB、8 KB
- ブロックごとのページ: 32、64、128、256、384、512
- 誤り訂正符号 (ECC) セクターサイズは、512 バイト (4、8、16 ビット訂正)、または 1024 バイト (24 ビット訂正) にプログラム可能です。

サポートされている NAND デバイスに関しては、次の表を参照ください。

表 4. サポートされている NAND デバイス

製品番号	製造元	容量	電圧	サポートカテゴリ
MT29F8G16ABBCAH4-IT	Micron	8 GB	1.8 V	動作確認済み

### 2.3.5. UART インターフェイスのデザイン・ガイドライン

HPS ブート・ファームウェアは、ブートプロセス全体を通してコンソール・ステータス・メッセージを HPS UART ポートに出力します。ブート・ファームウェア・コンソール出力を確認する場合は、次のガイドラインを考慮し、HPS UART ペリフェラルを HPS ブート時に使用可能なデバイス I/O に割り当てます。

**ガイドライン:** HPS 先行のブートおよびコンフィグレーション・スキームでは、HPS UART ペリフェラルを HPS 専用 I/O バンクに割り当てます。

SDM は、HPS を起動する前に HPS 専用 I/O および HPS SDRAM I/O をコンフィグレーションし、ユーザーモードにリリースします (Early I/O Release フロー)。そのほかの FPGA I/O およびファブリックは、ブートフローの後半で残りの FPGA がコンフィグレーションされるまで使用できません。

**ガイドライン:** FPGA 先行のブートおよびコンフィグレーション・スキームでは、HPS UART を HPS 専用 I/O もしくは FPGA I/O に割り当てることができます。

SDM は HPS を起動する前に、I/O リング全体を含む FPGA 部分全体をコンフィグレーションします。

**ガイドライン:** FPGA ファブリックを介して UART 信号をルーティングする場合、フロー制御信号を適切に接続します。

FPGA を介して UART 信号をルーティングする場合、フロー制御信号を利用できます。フロー制御を使用しない場合、FPGA で信号を次の表に示すように接続します。

表 5. UART インターフェイス・デザイン

信号	方向	接続
CTS	入力	Low
DSR	入力	High
DCD	入力	High
RI	入力	High
DTR	出力	接続なし
RTS	出力	接続なし
OUT1_N	出力	接続なし
OUT2_N	出力	接続なし

詳細は *Intel Stratix 10 Hard Processor System Technical Reference Manual* の「UART Controller」の章を参照ください。

#### 関連情報

- [Intel Stratix 10 Hard Processor System Technical Reference Manual](#)
- [Intel Stratix 10 SoC Boot User Guide](#)

### 2.3.6. I<sup>2</sup>C インターフェイスのデザイン・ガイドライン

**ガイドライン:** FPGA ファブリックを介して I<sup>2</sup>C 信号をルーティングする際は、オープンドレイン・バッファをインスタンス化します。

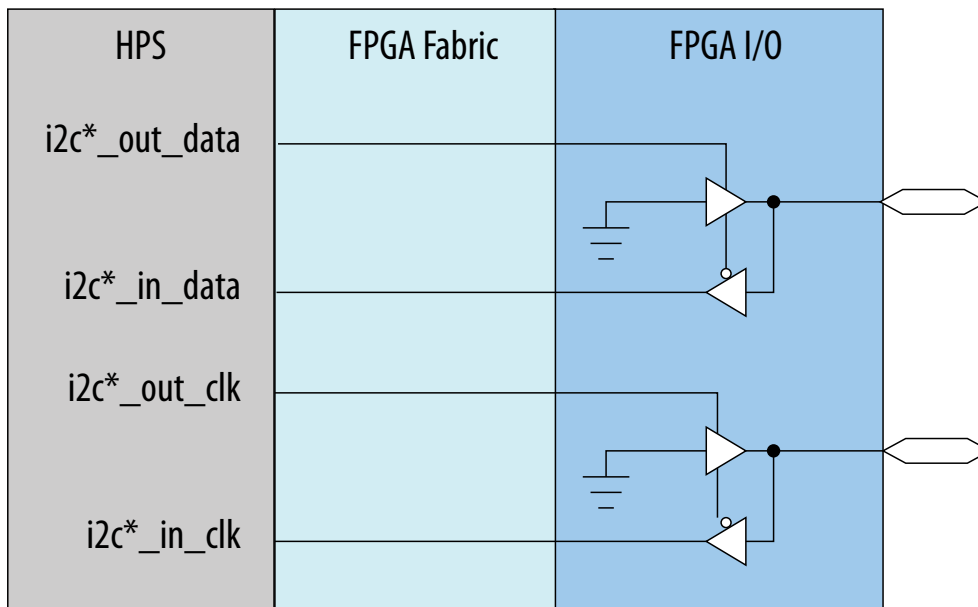
FPGA を介し I<sup>2</sup>C 信号をルーティングする場合、HPS から FPGA ファブリックへの I<sup>2</sup>C ピン (`i2c*_out_data`、`i2c*_out_clk`) はオープンドレインではなく、論理レベルが反転していることに注意してください。よって、論理レベル 0 を I<sup>2</sup>C バスに駆動するには、対応するピンを High に駆動します。この実装は、それらをトライステート・バッファの出力イネーブルに直接結び付けるために使用できるので便利です。オープンドレイン・バッファを実装するには、`altiobuff` を使用する必要があります。

**ガイドライン:** プルアップがボードデザインの外部 SDA および SCL 信号に追加されていることを確認します。

I<sup>2</sup>C 信号はオープンドレインであるため、バス上のデバイスがバスを Low に引き下げている際に、バスを確実に High に引き上げるためにプルアップが必要です。



図 -9: FPGA ピンへの I<sup>2</sup>C 配線



ガイドライン: High および Low のクロックカウントが、I<sup>2</sup>C インターフェイスの速度に対して正確にコンフィグレーションされていることを確認します。

I<sup>2</sup>C 内部クロックは以下の位置にあります。

- SDM—125 MHz
- HPS—100 Mhz

High および Low のクロックカウントのデフォルトの設定は 125 MHz にコンフィグレーションされているため、HPS I<sup>2</sup>C のデフォルトの High および Low クロック想定よりも長くなります。

## 2.4. HPS EMIF デザインの考慮事項

HPS の重要なコンポーネントは、外部 SDRAM メモリーです。以下に示すデザインに関する考慮事項は、SDRAM メモリーと HPS 間のインターフェイスを適切にデザインするためのものです。

外部 SDRAM を HPS に接続する際は、次の EMIF プランニング・ツールと必須ドキュメントを参照してください。

### EMIF プランニング・ツール

ツール	説明
インテル® FPGA 向け外部メモリー・インターフェイス IP - サポート・センター	インテル® FPGA 向け外部メモリー・インターフェイス IP - サポート・センターは、インテル FPGA に向けた外部メモリー・インターフェイスのデザインを支援する一連のツールおよびドキュメント・リソースです。
EMIF デバイスセクター	これは、アプリケーションの各外部メモリー・インターフェイスの種類およびパフォーマンス特性に基づき、インテル Stratix 10 SoC デバイスパッケージのリストを短時間で特定するための使い勝手の良いツールです。このツールは、リスト内の各デバイスパッケージの残りの汎用 I/O およびトランシーバー数を報告します。
EMIF スペック・エスティメーター	これは、必要な SoC デバイスの速度グレードを決定するための使いやすいツールです。アプリケーションのメモリー・インターフェイスの数、タイプおよびパフォーマンス特性を実装できるデバイスパッケージを特定後にこのツールを使用します。

EMIF IP の生成および、インテル Quartus Prime のコンパイルとタイミング・クロージャー支援ツールに関しては、インテル FPGA 向け外部メモリー・インターフェイス IP - サポート・センターを参照ください。

### 必須ドキュメント

ドキュメント	説明
<i>Intel Stratix 10 General Purpose I/O User Guide</i>	<i>Intel Stratix 10 General Purpose I/O User Guide</i> は、I/O カラムのアーキテクチャーと、HPS にアクセス可能な特定のハード・メモリー・コントローラー・ブロックの位置について説明しています。 HPS にアクセス可能なハード・メモリー・コントローラー・ブロックを HPS に接続するためのガイダンスについては、 <i>General Purpose I/O User Guide</i> の <b>Section 1.3: Modular I/O Banks Location and Pin Counts in Stratix 10 Devices</b> を参照ください。この章は、インテル Stratix 10 ファミリーのすべてのバリエーションにおけるデバイスとパッケージの組み合わせすべてに対する I/O カラムとバンク位置を示しています。これには、アクセス可能なバンクに対する HPS の相対的な位置も含まれます。
<i>External Memory Interfaces Intel Stratix 10 FPGA IP User Guide</i>	Intel Stratix 10 External Memory Interfaces User Guide には、HPS 外部メモリー・インターフェイスに使用される特定の I/O バンクおよび、アドレスもしくはコマンド、ECC およびデータ信号の位置を理解するための詳細が含まれています。このユーザーガイドにはまた、それらの外部メモリー・インターフェイス信号のバンク内での配置における制約や、デフォルトの配置から変更することができる柔軟性についての重要な情報も含まれます。インテルでは、このユーザーガイドで提供されている内容すべてを十分に把握することを推奨していますが、アプリケーションで HPS IP に向けて インテル Stratix 10 EMIF を適切にデザインするためには、次の章を理解していることが前提となります。 <ul style="list-style-type: none"> <li>• <b>Section 5.3.3.1. General Guidelines</b>—この章では、インテル Stratix 10 SX デバイスとパッケージの組み合わせでサポートされるメモリーの種類数と幅を示します。</li> <li>• <b>Chapter 2: Intel Stratix 10 EMIF IP Product Architecture</b>—この章は、I/O カラム、HMC、I/O レーンおよび、I/O エLEMENTの DDR SDRAM メモリーに対する強化された機能サポートに関してより詳しく説明しています。</li> <li>• <b>Section 2.7.1: Restrictions on I/O Bank Usage for Intel Stratix 10 EMIF IP with HPS</b>—この章では、アドレスまたはコマンド、ECC、およびデータ信号の特定の I/O バンクとレーンの位置を示す図を提供します。</li> </ul>

次のデザイン・ガイドラインは、上記の参照ドキュメントに記載されている情報を補足します。

### 関連情報

- [Intel Stratix 10 General Purpose I/O User Guide](#)
- [インテル® FPGA 向け外部メモリー・インターフェイス IP - サポート・センター](#)



- External Memory Interfaces Intel Stratix 10 FPGA IP User Guide

### 2.4.1. HPS を SDRAM に接続するための考慮事項

インテル Stratix 10 HPS のハード・メモリー・コントローラーは、ほかのハード・メモリー・コントローラーとともに FPGA I/O カラムにあります。HPS は 1 つのハード・メモリー・コントローラーのみを使用でき、それは、アドレスまたはコマンド、および ECC 信号が属する I/O バンク 2M の HPS ブロックの最も近くに配置されます。I/O バンク 2N は、16 ビットおよび 32 ビットのインターフェイス DQ/DQS データグループ信号に使用します。I/O バンク 2L は、64 ビットのインターフェイス DQ/DQS データグループ信号に使用されます。

#### インテル Stratix 10 HPS EMIF IP のインスタンス化

外部 SDRAM を インテル Stratix 10 HPS に接続するには、HPS に固有の EMIF IP を使用する必要があります。次のガイドラインに従い、正しい EMIF IP を HPS に適切にインスタンス化およびコンフィグレーションしてください。

**ガイドライン: プラットフォーム・デザイナーで、インテル Stratix 10 外部メモリー・インターフェイスを HPS IP に向けてインスタンス化します。**

プラットフォーム・デザイナーで特定の EMIF IP を使用し、HPS を外部 SDRAM メモリーに接続する必要があります。

EMIF モジュールは IP カタログペインにあります。**Library > Processors and Peripherals > Hard Processor Components > External Memory Interfaces for HPS Intel Stratix 10** の順に選択してください。

**ガイドライン: hps\_emif コンジットを HPS コンポーネントに接続します。**

プラットフォーム・デザイナーで HPS を EMIF に接続するには、インスタンス化された **emif\_s10\_hps\_0** モジュールの hps\_emif コンジットを、**stratix10\_hps\_0 module** の hps\_emif コンジットに接続する必要があります。

**ガイドライン: デバイスのコンフィグレーションを開始する前に、フリーランニングで安定したリファレンス・クロック・ソースを外部メモリー・インターフェイスに提供する必要があります。**

詳細は、*Intel Stratix 10 External Memory Interfaces IP User Guide* を参照ください。

**ガイドライン: HPS が外部 SDRAM または L3 SDRAM インターコネクタのリソースにアクセスしている間、HPS EMIF IP ブロックがリセットされないようにしてください。**

アプリケーションが HPS EMIF IP のリセットアサートと連携してコンテキストを保存および回復できない限り、HPS EMIF IP ブロックへのリセットのアサートは、HPS のリセットアサートに一致する必要があります。これは、HPS EMIF リセット入力、HPS リセット出力 (h2f\_reset、h2f\_cold\_reset など) または、HPS コールドリセット入力もソースするシステム内のほかのリセット (nCONFIG および HPS\_COLD\_nRESET リセット入力ピンなど) ソースからのリセットの 1 つまたは組み合わせに接続するだけで実現できます。

上記で説明されている方法ではなく、HPS をリセットせずに HPS EMIF IP をリセットする場合、アプリケーションは HPS EMIF IP リセットがアサートされる前に、リセット・マネージャーの brgmodrst レジスター、ビット 6 (ddrsch) を使用して L3 SDRAM インターコネクタをリセットにする必要があります。また、HPS EMIF IOPLL がロックされるまでそれを維持し続ける必要があります。そうでなければ、外部 SDRAM、または L3 SDRAM インターコネクタのリソースへのその後のアクセスにおいてプロセッサがロックアップする可能性があります。

**ガイドライン:** HPS メモリー・コントローラーのデータ・マスク (DM) ピンが有効になっていることを確認します。

プラットフォーム・デザイナーでメモリー・コントローラーをインスタンス化する際は、データマスクピンを有効にするチェックボックスを選択する必要があります。このコントロールが有効になっていない場合、メモリーのネイティブ・ワード・サイズよりも小さい SDRAM のデータにマスターがアクセスするたびに、データの破損が発生します。

**注意:** データのマスクングを有効にするチェックボックスは、メモリーサブタブにある Topology セクション内の External Memory Interfaces for HPS Intel Stratix 10 Intel FPGA IP の「Parameters」タブにあります。

**ガイドライン:** HPS IP に向けた Stratix 10 EMIF および、ご利用の特定のデバイスとパッケージの組み合わせでサポートされているコンフィグレーションの DDR3 または DDR4 コンポーネント、もしくはモジュールのみを選択します。

インテルの外部メモリー・インターフェイス・スペック・エスティメーターは、インテル FPGA および SoC デバイスでサポートされている外部メモリー・インターフェイスの種類、コンフィグレーションおよび最大のパフォーマンス特性を比較できるパラメトリック・ツールです。

#### 関連情報

- [外部メモリー・インターフェイスのウェブページ](#)
- [Intel Stratix 10 External Memory Interfaces IP User Guide](#)

### 2.4.2. HPS SDRAM I/O の位置

HPS IP に向けた インテル Stratix 10 EMIF は、すべての外部メモリー・インターフェイス信号に対するデフォルトのピン位置の割り当てを制約ファイル内に含みます。これは IP 生成時に作成され、インテル Quartus Prime 開発ソフトウェア・プロ・エディションはデザインのコンパイル時にそれを読み取ります。

**ガイドライン:** インテルでは、自動的に生成されるこれらのデフォルトのピン位置割り当てを開始点として使用することを推奨しています。

場合によっては、このセクションに示されている制約を満たすために、デフォルトのピンアウトを変更する必要があります。

**ガイドライン:** ボードレイアウトを確定する前に、「output\_files」サブフォルダーにある インテル Quartus Prime プロジェクトのピンアウトファイルで、HPS メモリー・コントローラー I/O の位置を確認します。

インテル Quartus Prime はデフォルトで、出力レポート、ログファイルおよびプログラミング・ファイルをプロジェクト・フォルダーの「output\_files」サブフォルダーに生成します。HPS EMIF のピン位置を含むデザインのピンアウトは、コンパイル後に .pin テキストファイルを確認します。

**ガイドライン:** HPS メモリー・インターフェイスに関連する I/O がすべて、アクティブな HPS EMIF I/O バンク内にあることを確認します。

機能している HPS メモリー・インターフェイスに必要な I/O がすべて、次の表のとおり HPS メモリー幅のアクティブなバンク内に位置していることが重要です。



表 6. HPS SDRAM I/O の位置

EMIF 幅	バンク 2N のレーン				バンク 2M のレーン				バンク 2L のレーン			
	3	2	1	0	3	2	1	0	3	2	1	0
16 ビット	GPIO		16 ビットデータ		NC <sup>(4)</sup>	アドレス/コマンド/RZQ/RefClk			GPIO			
16 ビット + ECC	GPIO		16 ビットデータ + ECC			アドレス/コマンド/RZQ/RefClk			GPIO			
32 ビット	32 ビットデータ				NC	アドレス/コマンド/RZQ/RefClk			GPIO			
32 ビット + ECC	32 ビットデータ + ECC					アドレス/コマンド/RZQ/RefClk			GPIO			
64 ビット	64 ビットデータ				NC	アドレス/コマンド/RZQ/RefClk			64 ビットデータ			
64 ビット + ECC	64 ビットデータ + ECC					アドレス/コマンド/RZQ/RefClk			64 ビットデータ + ECC			

### ピンの割り当て

単一の x8 DQS グループを実装する単一のデータレーンでは以下が当てはまります。

- DQ ピンは、インデックス 1、2、3、6、7、8、9、10 のピンを使用する必要があります。最終的なピンアウトがこれらのインデックスのピンのみを使用している限り、DQ ビット間で位置を入れ替えることが可能です。つまり、DQ[0]と DQ[3]の位置を入れ替えることが可能です。
- DM/DBI ピンはインデックス 11 のピンを使用する必要があります。これを柔軟に変更することはできません。
- DQS はインデックス 4 のピンを使用し、DQS#はインデックス 5 のピンを使用する必要があります。これを柔軟に変更することはできません。
- ALERT#ピンは I/O バンク 2N、レーン 0、ピン・インデックス 0 または、I/O バンク 2N、レーン 1、ピン・インデックス 0 に配置します。それ以外は、許可される場合に HPS REFCLK\_P、アドレス、コマンド、RZQ、RefClk または汎用 I/O に使用される場合を除き、ピン・インデックス 0 は「接続なし」でなければなりません。
- データレーンの割り当ては、上記表に示されているとおりに行う必要があります。最終的なピンアウトが、上記表のとおり HPS EMIF コンフィグレーションで許可されるレーンのみを使用している限り、バイトレーン全体の位置を入れ替えることが可能です。つまり、バイト 0 とバイト 1 の位置を入れ替えることができます。
- I/O バンク 2M のレーン 0、1 および 2 はアドレス、コマンド、RZQ、RefClk にのみ使用し、それ以外の場合は「接続なし」でなければなりません。
- アドレスおよびコマンドピンの配置はデフォルトから変更することはできません。
- ECC を使用しない場合、IO バンク 2M のレーン 3 は「接続なし」にする必要があります。ECC を使用している場合、ECC DQS グループは、特に制限されていない単一のデータレーンのいずれかに配置できます。つまり、2M に配置される ECC DQS グループに対する要件はありません。
- HPS REFCLK\_P は IO バンク 2M のレーン 2、ピン・インデックス 0 を使用し、HPS REFCLK\_N は IO バンク 2M のレーン 2、ピン・インデックス 1 を使用する必要があります。
- RZQ は IO バンク 2M のレーン 2、ピン・インデックス 2 を使用する必要があります。

(4) NC は「接続なし」を意味します。

### DQ または DQS グループの配置

コンフィグレーション	DQS グループの配置
16 ビット	2N の I/O レーン 0 および 1 に配置する必要があります。
16 ビット + ECC	2N の I/O レーン 0 と 1、および 2M の I/O レーン 3 に配置する必要があります。
32 ビット	2N に配置する必要があります。
32 ビット + ECC	2N および 2M の I/O レーン 3 に配置する必要があります。
64 ビット	2N と 2L に配置する必要があります。
64 ビット + ECC	2N、2L および 2M の I/O レーン 3 に配置する必要があります。

**注意:** いずれの場合においても、DQ および DQS グループは、示されている I/O バンクで入れ替え可能です。2M に配置される ECC DQS グループに対する要件はありません。

#### 未使用の HPS EMIF I/O を FPGA GPIO として利用する

- バンク 2N (データ[31:0])—16 ビット・インターフェイスの未使用レーンは FPGA GPIO に利用可能です。
- バンド 2M (Addr/Cmd/ECC)
  - レーン 0、1、2 は FPGA GPIO として利用できません。
  - ECC を使用していない場合、レーン 3 は FPGA GPIO として利用できません。
- バンク 2L (データ[63:32])—32 ビット以下のインターフェイスの未使用レーンは FPGA GPIO として利用可能です。

## 2.5. HPS メモリーのデバッグ

**ガイドライン:FPGA EMIF と外部メモリー・ツールキットを使用し、メモリー・インターフェイスが動作することを確認します。**

HPS SDRAM コントローラーは外部メモリー・インターフェイス・ツールキットをサポートしていないため、まず HPS メモリー・コントローラー以外を使用しメモリー・インターフェイスが動作することを確認してください。FPGA メモリー・コントローラーをインスタンス化し、HPS メモリー・コントローラーが使用する I/O と同じ I/O にルーティングするデザインを作成します。インターフェイスが EMIF ツールキットで動作することを確認後、HPS IP に向けた インテル Stratix 10 外部メモリー・インターフェイスを適切にインスタンス化してください。インスタンス化は、*External Memory Interface Handbook Volume 3: Reference Material* の「Compiling Intel Stratix 10 EMIF IP with the Intel Quartus Prime Software」の章にあるサブセクション、Instantiating the Intel Stratix 10 EMIF IP の説明に従い行います。

詳細は、以下のドキュメントを参照ください。

- *External Memory Interface Handbook Volume 3: Reference Material*
- *External Memory Interfaces Intel Stratix 10 FPGA IP User Guide*
- インテル® FPGA デバイス用 ピンアウトファイル

#### 関連情報

- [External Memory Interfaces Intel Stratix 10 FPGA IP User Guide](#)
- [インテル® FPGA デバイス用 ピンアウトファイル](#)



- External Memory Interface Handbook Volume 3: Reference Material

## 2.6. HPS のバウンダリー・スキャン

HPS JTAG インターフェイスは、バウンダリー・スキャン・テスト (BST) をサポートしていません。HPS I/O でバウンダリー・スキャン・テストを実行するには、最初に FPGA JTAG と HPS JTAG を内部でチェーン接続し、FPGA JTAG からバウンダリー・スキャンを発行する必要があります。

**ガイドライン: FPGA JTAG インターフェイスと HPS JTAG インターフェイスを内部でチェーン接続し、バウンダリー・スキャン・テストを実行します。**

FPGA と HPS JTAG を内部でチェーン接続するには、Quartus の **Device and Pins Options** に移動し、**Configuration** カテゴリを選択します。**HPS debug access port (DAP)** 設定で、ドロップダウン・オプションから **SDM Pins** を選択します。バウンダリー・スキャンを使用しない場合、FPGA JTAG と HPS JTAG インターフェイスは個別に使用することができます。HPS 専用 I/O を HPS JTAG のインターフェイスとして選択するには、ドロップダウン・オプションから **HPS Pins** を選択します。

## 2.7. エンベデッド・ソフトウェアのデバッグとトレース

このデバイスには、FPGA JTAG および HPS JTAG の JTAG ポートが 1 つだけあります。FPGA JTAG および HPS JTAG はチェーン接続したり個別に使用したりできます。

**ガイドライン: インテル では、開発およびフィールド問題のデバッグや診断に使用することができる、ボードへの利用可能な JTAG 接続を備えることを推奨しています。**

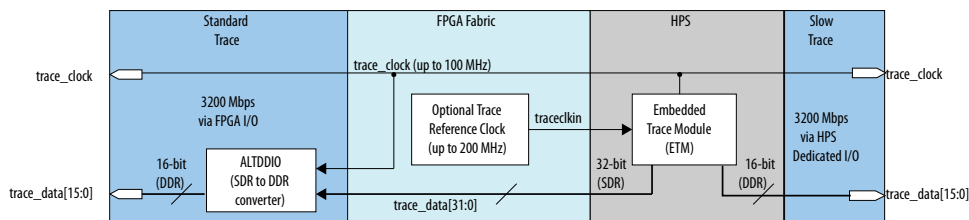
HPS は、2 つのトレース・インターフェイスを HPS 専用 I/O もしくは FPGA I/O を介して提供します。HPS 専用 I/O を介するインターフェイスは 16 ビット DDR インターフェイスで、低帯域幅トラフィックのトレースに使用できます (低周波数で動作する MPU など)。

トレース帯域幅を向上させるには、標準トレース・インターフェイスである FPGA への 32 ビット・シングル・データレート・インターフェイスを使用することができます。トレースモジュールは一般的にトレースデータがダブル・データレートで送信されることを想定しているため、シングル・データレートのトレースデータをダブル・データレートに変換する必要があります。

インテルでは、DDIO Megawizard IP をインスタンス化し、それを出力専用モードに設定してこの変換を実行することを推奨しています。トレースデータの最下位 16 ビットは最初にオフチップに送信される必要があるため、それらのビットを DDIO IP の `datain_l[15:0]` ポートに接続します。

トレースベンダーのデータシートを参照し、トレースパスに終端が必要かを判断してください。トレースベンダーが必要と示す終端が含まれない場合、トレースデータの破損につながる、またはインターフェイスの最大動作周波数が制限される可能性があります。

図 -10: トレース図



## 2.8. インテル Stratix 10 SoC FPGA のボード・デザイン・ガイドライン改訂履歴

表 7. Stratix 10 SoC FPGA のボード・デザイン・ガイドライン改訂履歴

ドキュメント・バージョン	変更内容
2019.04.17	RGMII を使用する際のタイミング問題のため、「HPS EMAC PHY インターフェイス」および「RGMII への適合」の章から RGMII のインスタンスを削除しました。
2019.02.27	リモート・システム・アップデート (RSU) の内容を追加しました。
2018.12.24	<ul style="list-style-type: none"> <li>PLL リファレンス・クロックおよび RZQ ピンを配置しなければならない位置に関する制約を含めて「HPS SDRAM I/O の位置」の章を更新しました。</li> <li>インテル Stratix 10 HPS EMIF のピン配置に関する必須規則を追加しました。</li> <li>次の章を削除しました。               <ul style="list-style-type: none"> <li>I/O バンク 2M、レーン 0、1、2 (Addr/Cmd)</li> <li>I/O バンク 2M、レーン 3 (ECC)</li> <li>I/O バンク、2N (データ)</li> <li>I/O バンク、2L (データ、64 または 72 ビット・インターフェイス)</li> </ul> </li> </ul>
2018.09.24	<ul style="list-style-type: none"> <li>ピンの名前を HPS_COLD_RESET から HPS_COLD_nRESET に変更しました。</li> <li>HPS_COLD_nRESET を使用されていない SDM I/O ピンにコンフィグレーションする手順を更新しました。</li> <li>「HPS クロック、リセット、PoR のピン機能と接続」の章にガイドラインを追加しました。HPS_COLD_nRESET をボード上のほかのリセットに接続しないでください。</li> </ul>
2018.05.07	<ul style="list-style-type: none"> <li>1000BASE-X PCS オプションの TSE MAC IP はトランシーバー I/O に向けたオプションを提供しなくなったため、FPGA トランシーバー I/O を使用し S10 HPS EMAC インスタンスに向けて SGMII PHY インターフェイスを実装するためのガイドラインを追加しました。</li> <li>HPS EMIF リファレンス・クロックが FPGA コンフィグレーションよりも先に安定することを反映するガイドラインを追加しました。</li> <li>LPDDR3 のインスタンスを削除しました</li> </ul>
2018.03.01	<ul style="list-style-type: none"> <li>「デバイス I/O を HPS ペリフェラルおよびメモリーに接続するためのデザインにおける考慮事項」の章にある SoC-FPGA I/O タイプの要約の表において、「SDM JTAG」を「SDM 専用ピンを介した FPGA JTAG」に置き換えました。</li> <li>HPS_COLD_nRESET の定義を訂正しました。これは「オープンドレイン」にはコンフィグレーションされません。</li> <li>「HPS のバウンダリー・スキャン」の章を更新しました。FPGA JTAG からのバウンダリー・スキャン発行方法および JTAG のチェーン接続方法を詳細に説明しています。</li> </ul>
2017.11.06	初版



### 3. Stratix 10 SoC FPGA の FPGA との接続

HPS と FPGA ファブリック間のメモリーマップによる接続は、デザインの最良のパフォーマンスを実現するための重要なツールです。推奨されるトポロジーに関するこの章のガイドラインを使用し、システムのパフォーマンスを最適化してください。

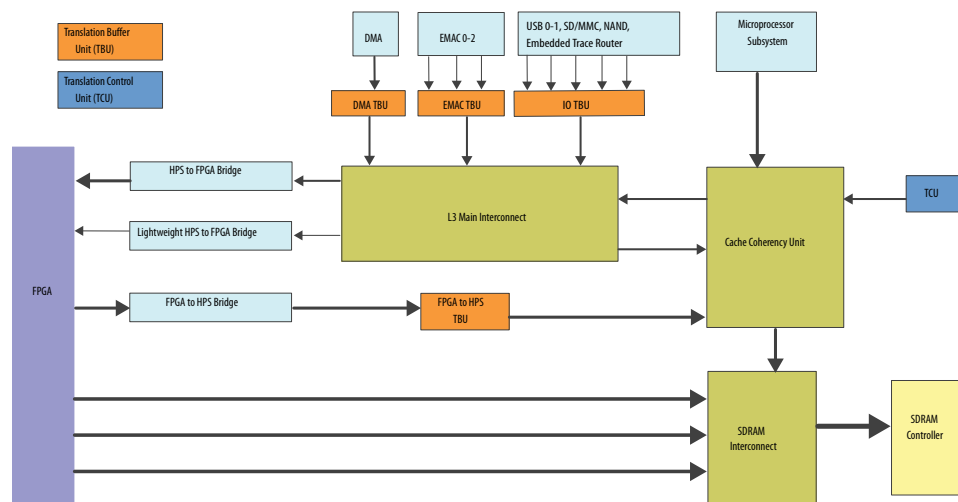
デザインの FPGA の残りの部分に関するデザイン・ガイドラインは、*Stratix 10 デバイスのデザイン・ガイドライン*で提供されています。

#### 3.1. HPS メモリーマップド・インターフェースの概要

HPS は HPS と FPGA 間の 3 つのメモリーマップド・インターフェースを公開します。

- HPS-to-FPGA ブリッジ: 32、64、または 128 ビット幅の Advanced Microcontroller Bus Architecture ( AMBA\* ) Advanced eXtensible Interface ( AXI\* )-4
- Lightweight HPS-to-FPGA ブリッジ: 32 ビット幅 AXI-4
- FPGA-to-HPS ブリッジ: 128 ビット幅 ACE\*-Lite
- FPGA-to-SDRAM AXI-4 ポート: 32、64、128 ビット幅の 3 つのインターフェースにより、HPS に接続する SDRAM に FPGA が直接アクセスすることを可能にします。

図 -11: Stratix 10 HPS の接続



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。

### 3.1.1. HPS-to-FPGA ブリッジ

**ガイドライン:** HPS-to-FPGA ブリッジを使用し、FPGA がホストするメモリーを HPS に接続します。

HPS-to-FPGA ブリッジにより、マイクロプロセッサ・ユニット (MPU)、DMA、または統合されたマスターを備えるペリフェラルなどの HPS のマスターは、SoC デバイスの FPGA 部分がホストをするメモリーにアクセスできます。このブリッジは 32、64、128 ビットのデータパスをサポートするため、ブリッジに接続された FPGA ファブリックの最大スレーブデータ幅に合わせて幅を調整できます。このブリッジは、バースト転送を実行するマスターが使用することを目的としており、FPGA ファブリックのペリフェラル・レジスターにアクセスするために使用することは避けてください。コントロールおよびステータス・レジスター・アクセスは、Lightweight HPS-to-FPGA ブリッジに送信します。

**ガイドライン:** HPS-to-FPGA ブリッジに接続されたメモリーが HPS ブートに使用される場合、SoC デバイスの FPGA 部分を最初にコンフィグレーションしてください。

HPS-to-FPGA ブリッジは、MPU を FPGA から起動する場合にアクセスされます。MPU を FPGA から起動する前に、SoC デバイスの FPGA 部分をコンフィグレーションし、HPS-to-FPGA ブリッジをアドレス指定可能なスペースに再マッピングする必要があります。これを行わないと、ブートプロセス中の HPS-to-FPGA ブリッジへのアクセスでバスエラーが発生します。これらの要件を満たすには、FPGA 先行のブートおよびコンフィグレーション・スキームを使用します。ブート・ファームウェア生成のための標準ツールフローは、HPS-to-FPGA ブリッジをアドレス指定可能なメモリー・スペースにマッピングします。

FPGA 先行のブートおよびコンフィグレーション・スキーム、Stratix 10 HPS に向けたブート・ファームウェア生成に関する詳細は、*Intel Stratix 10 SoC Boot User Guide* を参照ください。

#### 関連情報

[Intel Stratix 10 SoC Boot User Guide](#)

### 3.1.2. Lightweight HPS-to-FPGA ブリッジ

**ガイドライン:** Lightweight HPS-to-FPGA ブリッジを使用し、HPS で制御する必要がある IP を接続します。

Lightweight HPS-to-FPGA ブリッジにより HPS のマスターは、SoC デバイスの FPGA 部分のメモリーマップされたコントロール・スレーブ・ポートにアクセスできます。通常、HPS 内の MPU のみがこのブリッジにアクセスし、FPGA のペリフェラルへのコントロールおよびステータス・レジスター・アクセスを実行します。

**ガイドライン:** Lightweight HPS-to-FPGA ブリッジを FPGA メモリーに使用しないでください。代わりに、HPS-to-FPGA ブリッジをメモリーに使用します。

MPU がペリフェラル内のコントロールおよびステータス・レジスターにアクセスする場合、これらのトランザクションは通常、厳しく順序付けられます (ポストされません)。Lightweight HPS-to-FPGA ブリッジをレジスター・アクセス専用にする事で、バースト・トラフィックが代わりに HPS-to-FPGA ブリッジにルーティングされるため、アクセス時間が最小限に抑えられます。ほとんどの IP コアは 32 ビットのコントロールおよびステータス・レジスターを実装しているため、Lightweight HPS-to-FPGA ブリッジは、FPGA ファブリックへの固定 32 ビット幅の接続を備えています。ただしプラットフォーム・デザイナーは、FPGA 部分で生成されたインターコネクで、32 ビット以外の幅にトランザクションを適応させることができます。



### 3.1.3. FPGA-to-HPS ブリッジ

**ガイドライン: FPGA-to-HPS ブリッジを、FPGA のマスターから HPS へのキャッシュ・コヒーレンシーなメモリー・アクセスに使用します。**

FPGA-to-HPS ブリッジにより、FPGA ファブリックに実装されたマスターは、HPS 内のメモリーおよびペリフェラルにアクセスできます。このブリッジは、固定の 128 ビットのデータバスをサポートします。プラットフォーム・デザイナーは、生成されたインターコネクトで幅の狭いマスターに対しデータ幅を適応させることができます。

**ガイドライン: FPGA-to-HPS ブリッジは、キャッシュ・コヒーレンシーなメモリー・アクセスを ACE-Lite プロトコルでサポートします。**

FPGA マスターは、キャッシュ・コヒーレンシーなアクセスに ACE-Lite のキャッシュ信号拡張を使用する必要があります。

キャッシュ・コヒーレンシーなトランザクションのための ACE-Lite プロトコル拡張に関しては、Arm\* Developer ウェブサイトの AXI and ACE Protocol Specification を参照ください。

#### 関連情報

[AMBA AXI and ACE Protocol Specification](#)

### 3.1.4. FPGA-to-SDRAM ポート

**ガイドライン: FPGA のマスターから HPS SDRAM へのキャッシュ不可能なアクセスには、FPGAto-SDRAM ポートを使用します。**

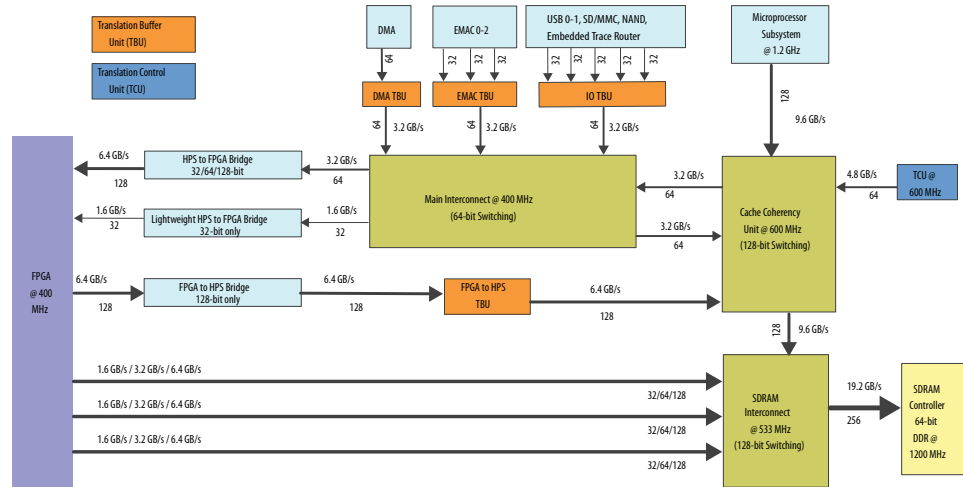
FPGA-to-SDRAM ポートにより、FPGA ファブリックに実装されたマスターは、CCU を経由するトランザクションなしに HPS SDRAM に直接アクセスできるようになります。FPGA-to-SDRAM ポートは 3 つあり (FPGA-to-SDRAM0, FPGA-to-SDRAM1, FPGA-to-SDRAM2)、32、64、128 ビットのデータバスをサポートしています。これらのインターフェイスは HPS SDRAM サブシステムにのみ接続するため、インテルでは、FPGA が HPS SDRAM への高スループットで低レイテンシーなアクセスを必要とする場合に、デザインにそれらを使用することを推奨しています。FPGA がキャッシュ・コヒーレンシーなアクセスを必要とし、ACE-Lite プロトコルでキャッシュ・コヒーレンシーなアクセスをサポートする FPGA-to-HPS ブリッジを使用する場合は、この推奨事項は当てはまりません。

### 3.1.5. インターフェイス帯域幅

HPS と FPGA ファブリック間でデータを移動するために使用するインターフェイスを特定するには、各インターフェイスの帯域幅を理解する必要があります。次の図は、HPS と FPGA ファブリック間で利用可能なピーク・スループットおよび HPS の内部帯域幅を表しています。ここに示す例では、FPGA ファブリックは 400 MHz、MPU は 1200 MHz、そして 64 ビットの外部 SDRAM は 1067 MHz で動作することを想定しています。

図 -12: Stratix 10 HPS のメモリーマップド帯域幅

略語については HPS メモリーマップド・インターフェイスの概要の図を参照ください。



各 HPS インターフェイスの相対遅延とスループット

インターフェイス	トランザクションの使用例	レイテンシー	スループット
HPS-to-FPGA	MPU が FPGA のメモリーにアクセスする場合	中	中
HPS-to-FPGA	MPU が FPGA のペリフェラルにアクセスする場合	中	非常に低い
Lightweight HPS-to-FPGA	MPU が FPGA のレジスターにアクセスする場合	低	低
Lightweight HPS-to-FPGA	MPU が FPGA のメモリーにアクセスする場合	低	非常に低い
FPGA-to-HPS	FPGA マスターがキャッシュ・コヒーレントではない SDRAM にアクセスする場合	高	中
FPGA-to-HPS	FPGA マスターが HPS オンチップ RAM にアクセスする場合	低	高
FPGA-to-HPS	FPGA マスターが HPS ペリフェラルにアクセスする場合	低	低
FPGA-to-HPS	FPGA マスターがコヒーレント・メモリーにアクセスしキャッシュミスになる場合	高	中
FPGA-to-HPS	FPGA マスターがコヒーレント・メモリーにアクセスしキャッシュヒットになる場合	低	中から高
FPGA-to-SDRAM	FPGA マスターが単一の FPGA-to-SDRAM ポートを介して SDRAM にアクセスする場合	中	高
FPGA-to-SDRAM	FPGA マスターが複数の FPGA-to-SDRAM ポートを介して SDRAM にアクセスする場合	中	非常に高い

注意: コンフィグレーションが推奨されていないインターフェイスに関しては、対応する「HPS-to-FPGAブリッジ」、「Lightweight HPS-to-FPGAブリッジ」、「FPGA-to-HPSブリッジ」インターフェイスの章を参照ください。



**ガイドライン: HPS-to-FPGA ブリッジを使用し、MPU から FPGA のペリフェラル・レジスターにアクセスすることは避けてください。**

HPS-to-FPGA ブリッジはトラフィックのバーストに最適化されており、ペリフェラル・アクセスは通常、1 ビートのみの短いワードサイズのアクセスです。そのため、HPS-to-FPGA ブリッジを介してペリフェラルにアクセスした場合、すでにインフライト状態のほかのバースト・トラフィックによってトランザクションがストールする可能性があります。

**ガイドライン: Lightweight HPS-to-FPGA ブリッジを使用し、MPU から FPGA のメモリーにアクセスすることは避けてください。**

Lightweight HPS-to-FPGA ブリッジは、非バーストのトラフィックに向け最適化されており、メモリーアクセスは通常、バーストとして実行されます (キャッシュ動作のために多くの場合 32 バイトです)。そのため、Lightweight HPS-to-FPGA ブリッジを介してメモリーにアクセスすると、スループットが制限されます。

**ガイドライン: FPGA-to-HPS ブリッジを使用し、非キャッシュ・コヒーレントの SDRAM に FPGA のマスターからアクセスすることは避けてください。**

FPGA-to-HPS ブリッジは、SDRAM 以外のアクセス (ペリフェラル、オンチップ RAM) に向け最適化されています。そのため、非コヒーレントなアクセスを実行して SDRAM に直接アクセスすると、FPGA-to-SDRAM ポートへのアクセスに比べてレイテンシーが増加し、スループットが制限されます。

**ガイドライン: FPGA のソフトロジック (DMA コントローラーなど) を使用し、共有データを HPS と FPGA 間で移動します。MPU および HPS DMA コントローラーをこの目的で使用することは避けてください。**

HPS と FPGA 間で共有データを移動する場合、インテルでは、MPU または HPS DMA コントローラーを使用してデータを移動するのではなく、FPGA から行うことを推奨しています。FPGA がキャッシュ・コヒーレントのデータにアクセスする必要がある場合は、適切な ACE-Lite キャッシュ拡張信号で FPGA-to-HPS ブリッジにアクセスし、キャッシュ可能なトランザクションを発行する必要があります。キャッシュ・コヒーレントではないデータを FPGA または HPS に移動する必要がある場合は、FPGA ロジックに実装されている DMA エンジンで FPGA-to-SDRAM ブリッジポートの 1 つを介してデータを移動することにより、可能な限り最高のスループットを実現します。HPS 内には、HPS と FPGA の間でデータを移動できる DMA エンジンが含まれていますが、その目的は、メモリーをマスターしないペリフェラルの支援、またはメモリー間のデータ移動を MPU に代わって提供することです。

#### 関連情報

- [HPS-to-FPGA ブリッジ \(34 ページ\)](#)
- [Lightweight HPS-to-FPGA ブリッジ \(34 ページ\)](#)
- [FPGA-to-HPS ブリッジ \(35 ページ\)](#)

## 3.2. 推奨されるシステムトポロジー

適切なシステムトポロジーを選択することは、可能な限り最高のスループットをデザインで実現することにつながります。最適なパフォーマンスを得るため、HPS と FPGA 間でデータを移動するためのインテルのトポロジー・ガイドラインに従ってください。これらのガイドラインは、キャッシュ・コヒーレントおよび非キャッシュ・コヒーレントのデータ移動をどちらも網羅しています。

### 3.2.1. FPGA ファブリックへの HPS のアクセス

HPS のマスターが FPGA ファブリックにアクセスするために使用できるブリッジは 2 つあります。それぞれのブリッジは特定のトラフィック・パターンに対して最適化されているため、HPS マスターが FPGA ファブリックにアクセスする必要がある場合、どちらのブリッジがシステムに適切かを決定する必要があります。

**ガイドライン: Lightweight HPS-to-FPGA ブリッジを介し、HPS を FPGA のソフト・ロジック・ペリフェラルに接続します。**

ハードウェア・デザインに HPS からアクセス可能なペリフェラルがある場合、それらを Lightweight HPS-to-FPGA ブリッジに接続する必要があります。ペリフェラルは通常、厳密に順序付けられた (ポストされない) アクセスによって 1 レジスターずつ HPS MPU からアクセスされます。アクセスは厳密に順序付けられているため、MPU からのトランザクションはスレーブからの応答が返されるまで完了しません。そのため、厳しく順序付けられたアクセスはレイテンシーに影響されやすくなります。よって、Lightweight HPS-to-FPGA ブリッジを HPS に含むことで、厳密に順序付けられたアクセスのレイテンシーを低減させます。

**ガイドライン: HPS は、HPS-to-FPGA ブリッジを介し FPGA メモリーに接続します。**

ハードウェア・デザインに HPS からアクセス可能なメモリーがある場合、それを HPS-to-FPGA ブリッジに接続する必要があります。Lightweight HPS-to-FPGA ブリッジとは異なり、HPS-to-FPGA ブリッジは、DMA 転送や FPGA メモリーからの MPU ソフトウェア実行などのバースト・トラフィックに使用することを目的としています。

**ガイドライン: HPS が FPGA ロジックのメモリーおよびペリフェラルにアクセスする必要がある場合、HPS-to-FPGA ブリッジと Lightweight HPS-to-FPGA ブリッジを使用します。**

FPGA ロジックに HPS からアクセス可能なメモリーとペリフェラルが混在する場合、HPS-to-FPGA ブリッジと Lightweight HPS-to-FPGA ブリッジをどちらもデザインに含めることが重要です。ペリフェラル・アクセスは通常レイテンシーに影響されやすいため、これらのアクセスに Lightweight HPS-to-FPGA ブリッジを使用することで、ほかのバーストアクセスが HPS-to-FPGA ブリッジを介して FPGA ファブリックに行われる際にスタベーションを防ぎます。複数の HPS マスターが FPGA ファブリックに同時にアクセスしている場合、両方のブリッジには並行してアクセス可能なため、ブリッジをどちらも含めることはシステムのパフォーマンス向上にもつながります。

### 3.2.2. FPGA とデータを共有する MPU

データのスルーputは、HPS と FPGA 間においてデータを共有する正しい方法を選択することで最適化することができます。この章では、HPS SDRAM がデータソースであり、FPGA がそれにアクセスする必要があると仮定します。FPGA が HPS SDRAM を起源とするデータにアクセスする方法は、主に 3 つあります。

- FPGA が FPGA-to-SDRAM ポートを介して直接データにアクセスする
- FPGA が FPGA-to-HPS ブリッジを介して直接データにアクセスする
- FPGA が HPS DMA を介して FPGA に移されたデータのコピーにアクセスする (非推奨)

SDRAM のデータがデータの最新コピーである (ソフトウェア管理のコヒーレンシーがある) 場合、最高のスルーputでデータにアクセスする方法は、FPGA のマスターが FPGA-to-SDRAM ポートを介してデータに直接アクセスすることです。



SDRAM のデータがデータの最新のコピーではない可能性があり、ソフトウェアが MPU キャッシュをフラッシュしてシステム全体のコヒーレンシーを維持しない場合、FPGA マスターが FPGA-to-HPS ブリッジにキャッシュ可能なトランザクションを実行し、確実に最新のデータにアクセスするようにします。

**HPS DMA コントローラーを FPGA と HPS 間のデータ移動に使用することは避けてください。代わりに FPGA ファブリックのソフト DMA コントローラーを使用します。HPS DMA コントローラーは、HPS 内にとどまるメモリーコピーまたはペリフェラルデータの移動にのみ使用します。**

HPS DMA を使用しデータを FPGA に移動させることは、HPS SDRAM への DMA 帯域幅が制限されているため非推奨です。HPS DMA は、MPU に代わってバッファを移動させるために使用する、もしくはペリフェラルとメモリー間の転送に使用することを目的としています。そのため、FPGA が HPS メモリーのバッファにアクセスする必要がある場合、もしくは HPS が FPGA に保存されているデータにアクセスする必要がある場合は、HPS がこれらの転送を開始するのではなく、FPGA のマスターがこれらの転送を実行することが常に推奨されます。

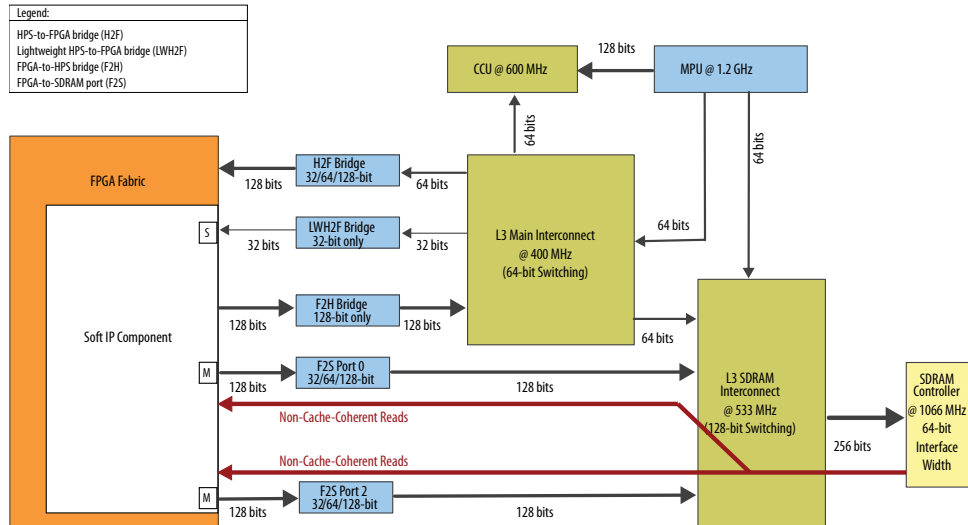
### 3.2.3. FPGA からのキャッシュ可能およびキャッシュ不可能なデータアクセスの例

#### 3.2.3.1. 例 1: FPGA で HPS SDRAM から直接データを読み取る

MPU がアクセスするものと同じデータのコピーに FPGA がアクセスするには、L1 データキャッシュと L2 キャッシュにデータのコピーが既にある場合はそれらをフラッシュする必要があります。HPS SDRAM にデータの最新コピーが含まれている場合に FPGA がこのデータにアクセスする最適なパスは、FPGA マスターが FPGA-to-SDRAM ポートを介してデータを読み取ることです。

図 -13: FPGA での HPS FPGA-to-SDRAM ポートからのデータ読み取り

この図は、128 ビット幅にコンフィグレーションされた 3 つの F2S ポートのうち 2 つを使用する例を表しています。



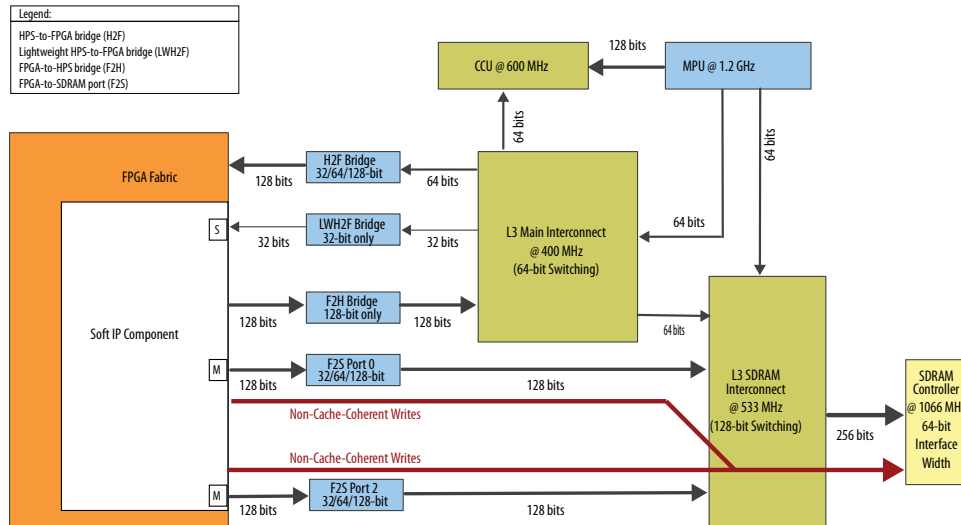
Stratix 10 HPS は、SDRAM への 128 ビットポートを最大 3 つサポートするため、FPGA に最大 3 つのマスターを実装し、各ポートを介して SDRAM のデータにアクセスすることで読み取りスループットを最大化できます。FPGA-to-SDRAM ポートを介して SDRAM へのパスを複数実装する場合、各ポートはそれぞれ独立して処理されるため、システムレベルで同期を行うようにしてください。1 つのポートをほかのポートよりも優先する場合、各ポートの QoS 設定を調整し、アプリケーションの必要性に応じてトラフィックパターンを形成できます。インテル では、バースト対応可能な FPGA のマスターを使用し、4 ビット以上のバースト長をポストできる FPGA-to-SDRAM ポートから読み取ることを推奨しています。(5)

### 3.2.3.2. 例 2: FPGA で HPS SDRAM に直接データを書き込む

この例では、HPS MPU は FPGA 内を起源とするデータへのアクセスを必要とします。データが書き込まれた後に MPU がコヒーレンシーを持ってデータにアクセスするためには、転送が開始される前にソフトウェアがキャッシュラインをフラッシュまたは無効化し、書き込み後、確実に SDRAM に最新データが含まれるようにする必要があります。キャッシュ操作を実行しないと、1 つまたは複数のキャッシュラインが最終的にエビクションされ、FPGA マスターによって書き込まれたデータが上書きされる可能性があります。

図 -14: FPGA での HPS FPGA-to-SDRAM ポートへのデータ書き込み

この図は、128 ビット幅にコンフィグレーションされた 3 つの F2S ポートのうち 2 つを使用する例を表しています。



注意: FPGA が FPGA-to-SDRAM ポートからデータを読み取る場合の例 1: FPGA で HPS SDRAM から直接データを読み取る (39 ページ) と同様、HPS SDRAM への書き込みスループットは、複数の 128 ビット FPGA-to-SDRAM ポートを使用し、それぞれのポートを少なくとも 1 つの FPGA のマスターに接続することにより最大化することができます。

### 3.2.3.3. 例 3: FPGA でキャッシュ・コヒーレンシーなデータを HPS から読み取る

この例では、FPGA は HPS を起源とするデータへのアクセスを必要とします。HPS の MPU がこのデータに最近アクセスしたため、データがまだキャッシュに含まれている可能性があります。よって、FPGA がキャッシュされたデータにアクセスすることが最適だと考えられます。ダーティー・キャッシュ・ラインをフラッシュするソフトウェアのオーバーヘッドを回避するために、FPGA は FPGA-to-HPS ブリッジに対してキャッシュ・コヒーレンシーな読み取りを実行できます。読み取られるバッファは比較的小さいことが重要です。そうでない場合、L2 キャッシュはほとんどの転送で、SDRAM からのデータの読み取りをスラッシングする可能性があります。大規模なバッファの転送には、例 1 で示されているように、FPGA が FPGA-to-SDRAM ポートから直接データを読み取るほうがより適切です。

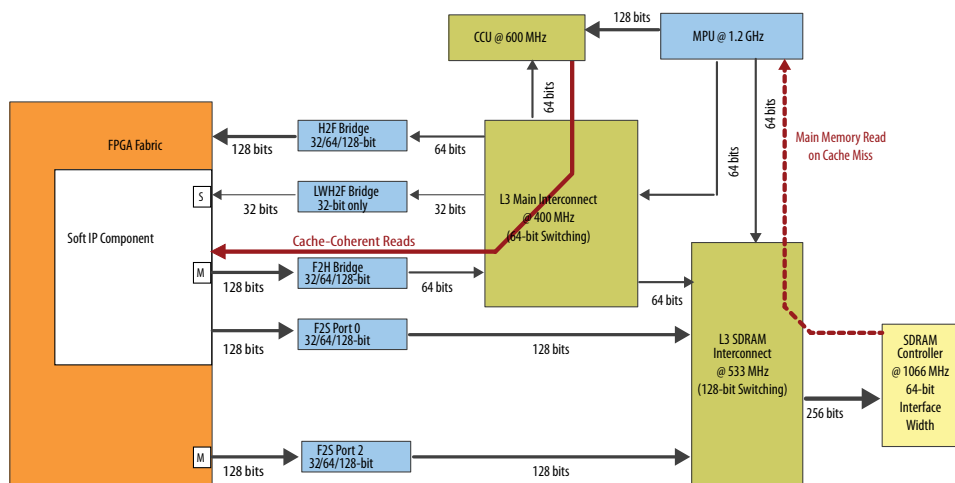
(5) HPS への Avalon® -MM バースト・トランザクションが、AXI プロトコルで指定されている 4 KB のアドレス境界制限を超えないことを確認してください。



**ガイドライン: FPGA-to-HPS ブリッジをターゲットとするフルアクセスを実行します。**

トランザクションをキャッシュ可能にするには、FPGA マスターが FPGA-to-HPS ブリッジから読み取りを行い、ACE-Lite プロトコルのキャッシュ拡張信号を利用する必要があります。キャッシュ・コヒーレンシーなアクセスのための ACE-Lite プロトコル信号拡張の詳細については、「関連情報」を参照ください。

図 -15: FPGA でのキャッシュ・コヒーレンシーなデータの読み取り



**ガイドライン: FPGA-to-HPS ブリッジをターゲットにし、64 バイトにアライメントされたキャッシュ可能なアクセスを実行します。**

HPS の CCU は、キャッシュラインと同じサイズ (64 バイト) のトランザクションに向けて最適化されています。そのため、データを 64 バイト境界に揃え、データ幅の調整後に、128 ビットの FPGA-to-HPS ブリッジポートへのバースト長が 4 ビットであることを確認する必要があります。例えば 64 ビットの FPGA マスターは、データを 64 バイトにアライメントし、バースト長 8 で 64 ビット全体のアクセスを実行する必要があります。

**ガイドライン: キャッシュ可能なトランザクションごとに 64 バイトにアクセスします。**

各バースト・トランザクションが 64 バイトにアクセスするようにします。各トランザクションは、64 バイト境界で開始する必要があります。

表 8. 64 バイト・アライメントのバースト長

FPGA マスター幅 (ビット)	アクセスサイズ (バイト)	バースト長
32	4	16
64	8	8
128	16	4

**関連情報**

AMBA AXI and ACE Protocol Specification

### 3.2.3.4. 例 4: FPGA でキャッシュ・コヒーレンシーなデータを HPS に書き込む

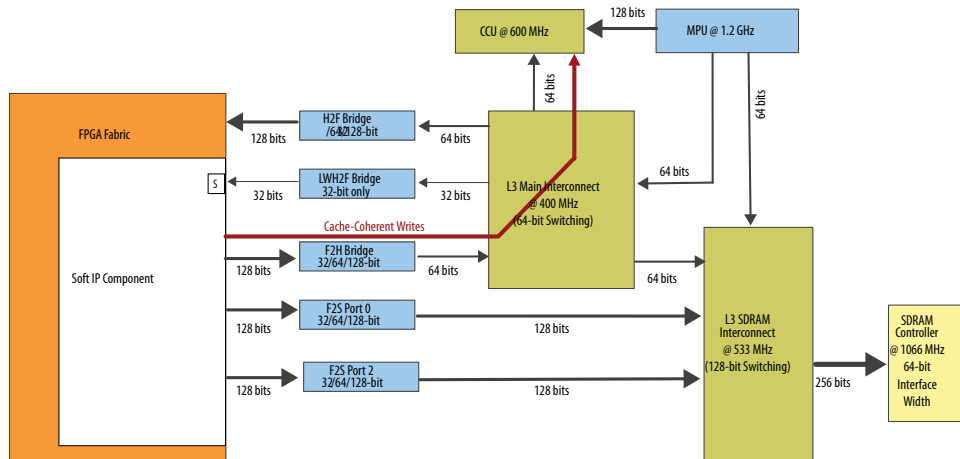
この例では、HPS MPU は FPGA を起源とするデータへのアクセスを必要とします。小さなデータブロックを MPU と共有する最も効率的なメカニズムは、FPGA 内のロジックに HPS へのキャッシュ可能な書き込みを実行させることです。HPS に書き込まれるデータは、比較的小さなブロックの形式であることが重要です。これは、大きなブロックの書き込みは L2 キャッシュのスラッシングにつながり、ほとんどの転送でキャッシュが SDRAM に書き込まれるようになるためです。大規模なバッファ転送においては、例 2 に示すように FPGA が FPGA-to-SDRAM ポートに直接データを書き込むほうが適切です。

**ガイドライン: FPGA-to-HPS ブリッジをターゲットとするフルアクセスを実行します。**

トランザクションをキャッシュ可能にするには、FPGA マスターが FPGA-to-HPS ブリッジに書き込み、ACE-Lite プロトコルのキャッシュ拡張信号を使用する必要があります。キャッシュ・コヒーレンシーなアクセスのための ACE-Lite プロトコル信号拡張に関する詳細は、関連情報を参照ください。

図 -16: FPGA でのキャッシュ・コヒーレンシーなデータの書き込み

略語については、HPS メモリーマップド・インターフェイスの概要にある図を参照ください。



**ガイドライン: FPGA-to-HPS ブリッジをターゲットとし、32 バイトにアライメントされたキャッシュ可能なアクセスを実行します。**

HPS の CCU スレーブは、キャッシュラインと同じサイズ (32 バイト) のトランザクションに向けて最適化されています。そのため、データを 32 バイト境界に揃え、データ幅の調整後に 64 ビットの CCU スレーブへのバースト長が 4 ビットであることを確認する必要があります。例えば、FPGA-to-HPS ブリッジが 128 ビットのトランザクションに向けて設定されている場合、データを 32 バイトに揃え、バースト長 2 で 128 ビット全体のアクセスを実行する必要があります。

**ガイドライン: L2 ECC が有効になっている場合、FPGA-to-HPS ブリッジへのキャッシュ可能なアクセスが 8 バイト境界でアライメントされていることを確認します。**

L2 キャッシュでエラー検出と訂正 (ECC) を有効にする場合はまた、8 バイトの各データグループが完全に書き込まれるようにする必要があります。L2 キャッシュは 64 ビット境界で ECC 動作を実行するため、キャッシュ可能なアクセスを実行する際にはかならずアクセスを 8 バイト境界に揃え、一度に 8 つのレーンすべてに書き込む必要があります。これらの規則に従わないとダブル・ビット・エラーが発生し、回復することができません。



ECC が有効になっているか無効になっているかにかかわらず、64 バイトのキャッシュ・トランザクションは最高のパフォーマンスをもたらします。64 バイトのキャッシュ・トランザクションの詳細については、例 3「FPGA でキャッシュ・コヒーレンシーなデータを HPS から読み取る」の章にある **ガイドライン: キャッシュ可能なトランザクションごとに 64 バイトにアクセスします** を参照してください。

**ガイドライン: L2 ECC が有効になっている場合、FPGA-to-HPS ブリッジへのキャッシュ可能なアクセスで 8 つの書き込みストロープのグループが有効になっていることを確認します。**

- 32 ビット FPGA マスターからの FPGA-to-HPS アクセスでは、書き込みバイトストロープをすべて有効にし、バースト長を 2、4、8 または 16 にする必要があります。
- 64 ビット FPGA マスターからの FPGA-to-HPS アクセスでは、すべての書き込みバイトストロープを有効にする必要があります。
- 128 ビット FPGA マスターからの FPGA-to-HPS アクセスでは、上位 8 または下位 8 (もしくは両方) の書き込みバイトストロープを有効にする必要があります。

#### 関連情報

- [AMBA AXI and ACE Protocol Specification](#)
- [例 3: FPGA でキャッシュ・コヒーレンシーなデータを HPS から読み取る \(40 ページ\)](#)

### 3.3. HPS-to-FPGA インターフェイス・デザインに推奨される開始点

ご利用のトポロジーに応じて、2 つのハードウェア・リファレンス・デザインのいずれかをご自身のハードウェア・デザインの開始点として選択できます。

**ガイドライン: インテルでは、HPS を FPGA のソフト IP に接続する例として、ゴールデン・ハードウェア・リファレンス・デザイン (GHRD) を開始点として使用することを推奨します。**

ゴールデン・ハードウェア・リファレンス・デザイン (GHRD) は、デザインを開始するにあたり、システムのベースとして使用するのに最適なデフォルト設定とタイミングを備えています。

詳細は、「ゴールデン・ハードウェア・リファレンス・デザイン (GHRD)」の章を参照してください。

#### 関連情報

[ゴールデン・ハードウェア・リファレンス・デザイン \(GHRD\) \(52 ページ\)](#)

### 3.4. FPGA アクセラレーターに向けたタイミング・クロージャ

FPGA に公開されている HPS ブリッジと FPGA-to-SDRAM ポートは同期しています。また、クロック・クロッシングはインターフェイス内で実行されます。そのため、FPGA に面しているロジックとユーザーデザインのタイミングが収束することのみを Timing Analyzer で確認してください。HPS は割り込みを非同期と見なすため、HPS ロジックはそれらを内部 HPS クロックドメインに再同期します。よって、割り込みのタイミングを収束する必要はありません。

### 3.5. ブリッジのコンフィグレーションおよび使用方法に関する情報

デフォルトで、SSBL はすべてのブリッジのリセット解除しか行わず、自動的にブリッジをコンフィグレーションしたり有効にしたりしません。すべてのブリッジはデザインに応じてコンフィグレーションし有効にする必要があります。これは、SSBL によって実行される "u-boot .scr" スクリプトファイルを作成す



ることで実現できます。ここで SSBL は、ブリッジのコンフィグレーションに必要なレジスターを変更します。この時点でブリッジはコンフィグレーションされ有効になり、その後の FPGA コンフィグレーション中に SSBL によって変更することはできません。

### 3.6. インテル Stratix 10 SoC FPGA の FPGA との接続の改訂履歴

表 9. Stratix 10 SoC FPGA の FPGA との接続の改訂履歴

ドキュメントバージョン	変更内容
2019.04.17	メンテナンス・リリース
2019.03.19	メンテナンス・リリース
2018.12.24	メンテナンス・リリース
2018.09.24	FPGA 先行のコンフィグレーション・モードと、HPS 先行のブートモードにおけるブリッジのコンフィグレーションおよび使用方法に関する情報を追加しました。
2018.05.07	メンテナンス・リリース
2018.03.01	メンテナンス・リリース
2017.11.06	初版

## 4. Stratix 10 SoC FPGA のシステムに関する考慮事項

### 4.1. タイミングに関する考慮事項

システムをコンフィグレーションする際に、以下の PHY インターフェイスを選択することができます。

- HPS EMAC PHY インターフェイス
  - Reduced Media Independent Interface (RMII)
  - Reduced Gigabit Media Independent Interface (RGMI)
- FPGA I/O を介して接続される PHY インターフェイス
  - GMII/MII
    - RMII—MII-to-RMII アダプターを使用
    - Serial Gigabit Media Independent Interface (SGMII)—GMII-to-SGMII アダプターを使用
  - インテル Management Data Input/Output (MDIO)

これら PHY インターフェイスのタイミングに関する考慮事項の詳細は、「HPS インターフェイスにおけるデザイン・ガイドライン」の章にある内容をそれぞれ参照してください。

#### 関連情報

[HPS インターフェイスにおけるデザイン・ガイドライン \(10 ページ\)](#)

#### 4.1.1. FPGA アクセラレーターに向けたタイミング・クロージャ

FPGA に公開されている HPS ブリッジと FPGA-to-SDRAM ポートは同期しています。また、クロック・クロッシングはインターフェイス内で実行されます。そのため、FPGA に面しているロジックとユーザーデザインのタイミングが収束することのみを Timing Analyzer で確認してください。HPS は割り込みを非同期と見なすため、HPS ロジックはそれらを内部 HPS クロックドメインに再同期します。よって、割り込みのタイミングを収束する必要はありません。

コンジットは、プラットフォーム・デザイナーでサポートされる標準インターフェイスに適合しない信号を伝えます。このコンジットには、FPGA ファブリックにルーティングされる HPS ペリフェラル外部インターフェイスや、HPS DMA ペリフェラル・リクエスト・インターフェイスなどがあります。

##### 4.1.1.1. HPS 先行のブートまたは FPGA 先行のブートにおける考慮事項

Stratix 10 SoC デバイスは、2 つのブートおよびコンフィグレーション・モードをサポートします。システムをデザインする際は、HPS 先行のブートモードまたは FPGA 先行のブートモードのいずれかをアプリケーションに選択する必要があります。

**注意:** HPS 先行のブートモードを使用すると、HPS のブート時間を短縮できます。

#### ガイドライン: エンジニアリング・サンプル・デバイスの制約

エンジニアリング・サンプル (ES) デバイスは、デュアルフラッシュを使用する HPS 先行ブートモードのみをサポートします。ES デバイスを使用し構築されるボードはすべて、HPS 先行のブートモードをサポートするようにデザインする必要があります。本番環境で別のブートスキームを使用する場合は、デュアルフラッシュまたはシングルフラッシュで、HPS 先行ブートおよび FPGA 先行ブートの両方にボードをデザインすることが可能です。

#### ガイドライン: HPS 先行のブートモードは Early I/O Release を使用します。

このドキュメントのガイドラインに従い、Early I/O Release に向けて HPS SDRAM インターフェイスのボードおよび SoC デバイスのピン配置を適切にデザインしてください。

サポートされているブートモードの詳細については、*Intel Stratix 10 SoC Boot User Guide* および *Intel Stratix 10 Hard Processor System Technical Reference Manual* の「Boot and Configuration」の章を参照ください。

#### 関連情報

- [Intel Stratix 10 Hard Processor System Technical Reference Manual](#)
- [Intel Stratix 10 SoC Boot User Guide](#)

### 4.1.2. USB インターフェイスのデザイン・ガイドライン

インテル Stratix 10 HPS では、1.8 V の専用 HPS I/O を使用し、組み込み USB MAC を業界標準の USB 2.0 ULPI PHY に直接接続することができます。FPGA 配線リソースを使用せず、タイミングが固定されているため、デザインは簡潔になります。

このガイドでは、サポートされているすべての PHY 動作速度 (高速 HS 480 Mbps、フルスピード FS 12 Mbps、および低速 LS 1.5 Mbps) を網羅するデザイン・ガイドラインについて説明します。

**ガイドライン: インテルでは、デバイスがクロックを供給する場合の USB PHY モードと、外部クロックがソースの場合の USB PHY モードをどちらもサポートするようにボードをデザインすることを推奨しています。**

Stratix 10 SoC 開発キットは Microchip USB3320 USB PHY を使用します。このデバイスは、HPS USB モジュールで動作することが知られています。

インテル Stratix 10 SoC の ULPI MAC と PHY 間のインターフェイスは、8 ビットのデータと次の制御信号で構成されます。

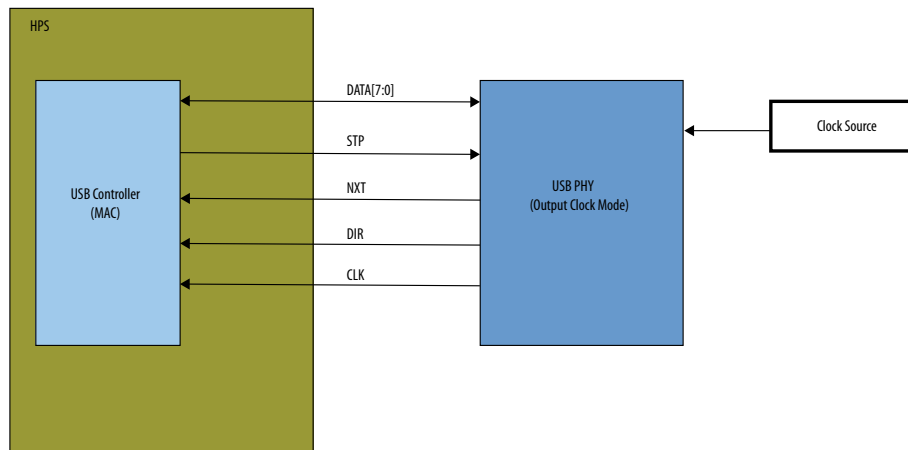
- STP
- NXT
- DIR

最後に、60 MHz の静的クロックが PHY または外部オシレーターから駆動され、一部の HPS から USB MAC のレジスターアクセスなどの動作に必要とされます。PHY メーカーより提供されているリセットおよび電源投入に関する推奨事項に従っていることを確認してください。

USB PHY が入力クロックモードと出力クロックモードをどちらもサポートする場合、インテルでは、ボードをどちらのモードもサポートするようにデザインし、潜在的なタイミング問題を軽減できるようにすることを推奨します。通常これらのモードは、High または Low にプルされる受動ブートストラップ・ピンで選択されます。

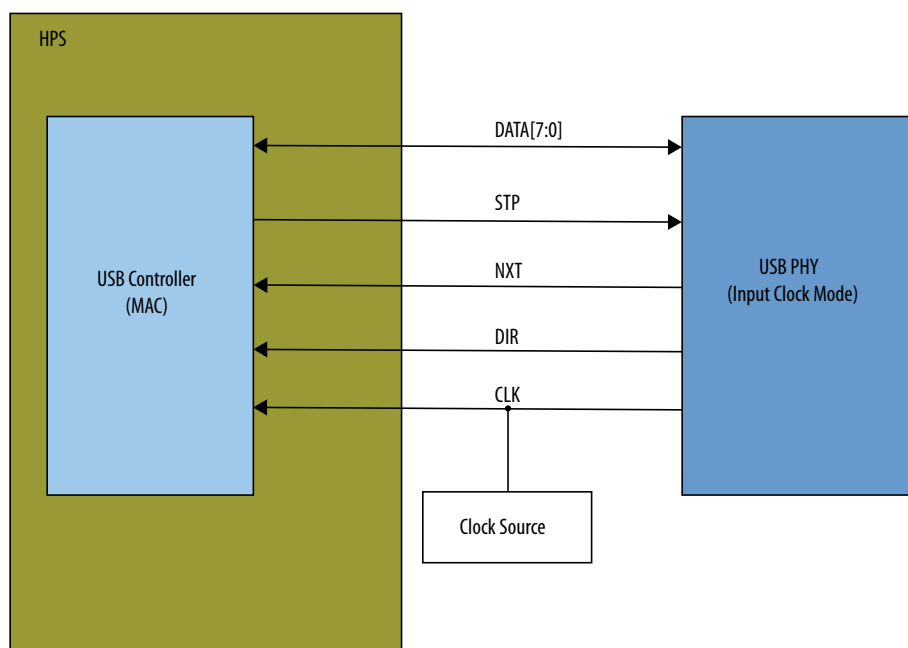
- 出力モード—出力クロックモードでは、クロックは USB PHY によって生成されます。信号はすべてこのクロックに同期しています。

図 -17: 出力モード



- 入力モード—入力クロックモードでは、PHY は外部ソースからのクロックを受信します。信号はすべてこのクロックに同期しています。このモードにおいてクロックは、FPGA の PLL または外部ソースで生成できます。

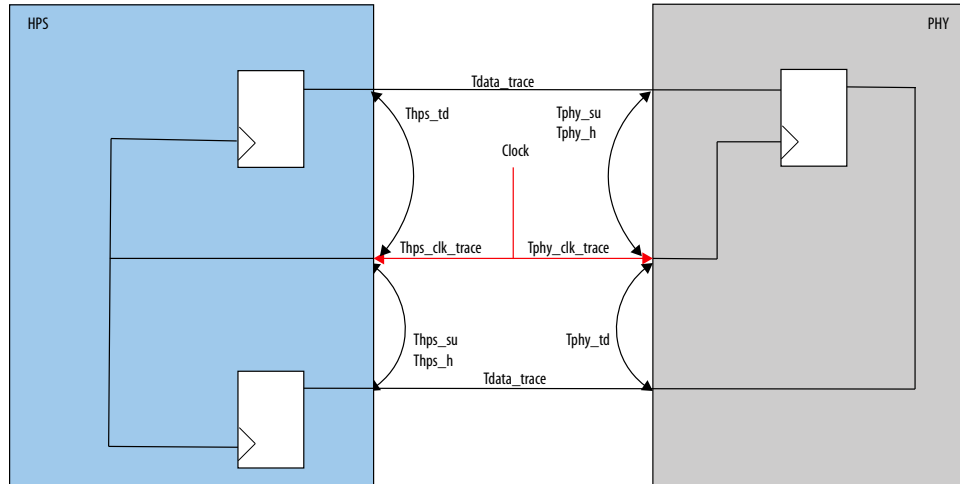
図 -18: 入力モード



**ガイドライン: USB 信号のトレース長が最小になっていることを確認します。**

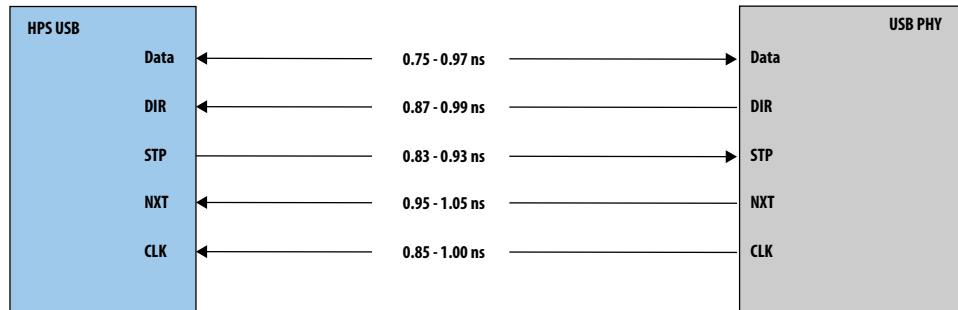
60 MHz における周期は 16.67 ns であり、その間に、例えばクロックは外部 PHY から MAC に移動し、そしてデータおよび制御信号は MAC から PHY に移動する必要があります。往復遅延があるため、クロックおよび ULPI 信号の最大長は重要です。予備のタイミングデータに基づき、最大長は 7 インチ未満にすることが推奨されます。これは 5 ns の  $T_{CO}$  仕様の PHY に基づいています。仕様がそれよりも遅い場合は、それに応じて全長を短くする必要があります。

図 -19: トレース長



バスの USB PHY 側にセットアップ・タイミングのマーヅンがほとんどない場合、PHY を入力クロックモードに切り替え、60 MHz のクロックソースをボードから供給することができる場合があります。

図 -20: システム図とボード仕様



**ガイドライン: シグナル・インテグリティーが考慮されていることを確認します。**

シグナル・インテグリティーは主に、PHY から HPS の MAC に駆動される CLK 信号において重要です。これらの信号は最大長のポイントツープイントであるため、通常は終端せずに行われますが、インテルでは、トレースをシミュレーションし反射を最小限に抑えることを推奨しています。シミュレーションで特に示されない限り、FPGA からの 50  $\Omega$  出力設定を使用することが一般的に推奨されます。可能であれば、PHY ベンダーが提供する同様の設定を使用します。





#### ガイドライン: OTG 動作を使用する場合はデザインを適切に行います。

On-the-Go (OTG) 機能を使用する場合、SoC はホストまたはエンドポイントになることができます。ホストモードにおいては、USB フラッシュドライブをサポートしている場合や、潜在的に USB ハードドライブをサポートしている場合などの電力供給を考慮してください。これらの電力要件と逆電流は通常、インテル Stratix 10 SoC FPGA 開発キットで使用されているような外部ダイオードと電流リミッターを使用し考慮する必要があります。

インテル Stratix 10 SoC 開発ボードの回路図に関しては、*Stratix 10 FPGA Development Kit User Guide* を参照ください。

## 4.2. 最大限のパフォーマンスの実現

HPS と FPGA ファブリック間のメモリーマップによる接続は、デザインのパフォーマンスを最大限に引き上げるための重要なツールです。

システムのパフォーマンスを最適にするために推奨されるトポロジーについては、「FPGA との接続」の章にあるガイドラインを参照してください。

#### 関連情報

[Stratix 10 SoC FPGA の FPGA との接続 \(33 ページ\)](#)

## 4.3. システムレベルのキャッシュ・コヒーレンシー

キャッシュ・コヒーレンシーは、システム内の複数のマスター間で常にデータを共有する必要があることを理解するための基本的な内容です。SoC デバイスにおけるこれらのマスターは、MPU、DMA、マスター・インターフェイスを備えるペリフェラル、および HPS に接続される FPGA のマスターになります。MPU にはレベル 1 およびレベル 2 のキャッシュ・コントローラーが含まれているため、システムのマインメモリーよりも多くの最新コンテンツを保持できます。HPS は、システム内のマスターがメモリーの一貫したビューをもつための 2 つのメカニズムをサポートします。つまり、メインメモリーに最新の値が含まれるようにする、または ACE-Lite インターフェイスを使用しマスターがディレクトリーベースの CCU ファブリックにアクセスするようにします。

MPU はバッファーをキャッシュ不可になるように割り当て、L1 および L2 キャッシュによってデータがキャッシュされないようにすることができます。MPU はまた、キャッシュ可能なデータにアクセスし、ほかのマスターがデータへのアクセスを試みる前にデータをメインメモリーにフラッシュするか、キャッシュ不可のバッファーにコピーすることができます。オペレーティング・システムは通常、キャッシュ・コヒーレンシーを維持する上記の両方の方法のメカニズムを提供します。

システム内のマスターは、MPU に依存しデータをキャッシュせずにメインメモリーに配置する、もしくはシステム内のマスターに CCU を介してキャッシュ可能なアクセスを実行させることにより、コヒーレンシーなデータにアクセスします。使用するメカニズムは、マスターがアクセスしているメモリーのバッファーサイズによって異なります。

詳細については、「FPGA との接続」の章を参照してください。

#### ガイドライン: スラッシングのオーバーヘッドを回避するため、CCU を介してアクセスされるデータが 1 MB の L2 キャッシュに収まるようにします。

L2 キャッシュのサイズは 1 MB であるため、合計サイズが 1 MB を超えるバッファーにシステム内のマスターが頻繁にアクセスするとスラッシングが発生します。

キャッシュのスラッシングは、データサイズがキャッシュサイズを超え、キャッシュが頻繁にエビクションとメインメモリーへのプリフェッチを引き起こす状況です。スラッシングは、データをキャッシュすることによるパフォーマンス上の利点を無効にします。

スラッシングの可能性がある状況では、マスターにキャッシュ以外のコヒーレンシーなデータにアクセスさせ、MPU で実行されるソフトウェアにシステム全体のデータ・コヒーレンシーを維持させる方法がより合理的です。

**ガイドライン: MPU とシステムマスターで共有されるデータの小さなバッファの場合、システムマスターにキャッシュ可能なアクセスを実行させ、キャッシュのフラッシュ動作によるオーバーヘッドを回避することを検討します。**

システム内のマスターがより小さなコヒーレント・データ・ブロックへのアクセスを必要とする場合、MPU がキャッシュ可能なメモリーとしてバッファにアクセスし、システム内のマスターがデータへのキャッシュ可能なアクセスを実行することを検討する必要があります。FPGA-to-HPS ブリッジでサポートされている ACE-Lite プロトコルでのキャッシュ可能な CCU へのアクセスは、マスターと MPU がデータの同じコピーにアクセスすることを保証しています。MPU がキャッシュ可能なバッファを使用し、システムマスターがキャッシュ可能なアクセスを実行することで、ソフトウェアは、MPU とシステムマスターがデータの同じコピーをもつようシステム全体のコヒーレンシーを維持する必要がなくなります。

**関連情報**

[Stratix 10 SoC FPGA の FPGA との接続 \(33 ページ\)](#)

## 4.4. Stratix 10 SoC FPGA のシステムに関する考慮事項の改訂履歴

表 10. Stratix 10 SoC FPGA のシステムに関する考慮事項の改訂履歴

ドキュメント・バージョン	変更内容
2019.04.17	RGMI2 を使用する際のタイミング問題のため、「タイミングに関する考慮事項」の章において RGMI2 のインスタンスを削除しました。
2019.03.19	メンテナンス・リリース
2018.12.24	メンテナンス・リリース
2018.09.24	メンテナンス・リリース
2018.05.07	メンテナンス・リリース
2018.03.01	メンテナンス・リリース
2017.11.06	初版

## 5. インテル Stratix 10 SoC FPGA 向けエンベデッド・ソフトウェアのデザイン・ガイドライン

### 5.1. 概要

この章では、インテル Stratix 10 ハード・プロセッサ・システムに向けてソフトウェア開発プラットフォームを構築するためのデザイン上の考慮事項について説明します。

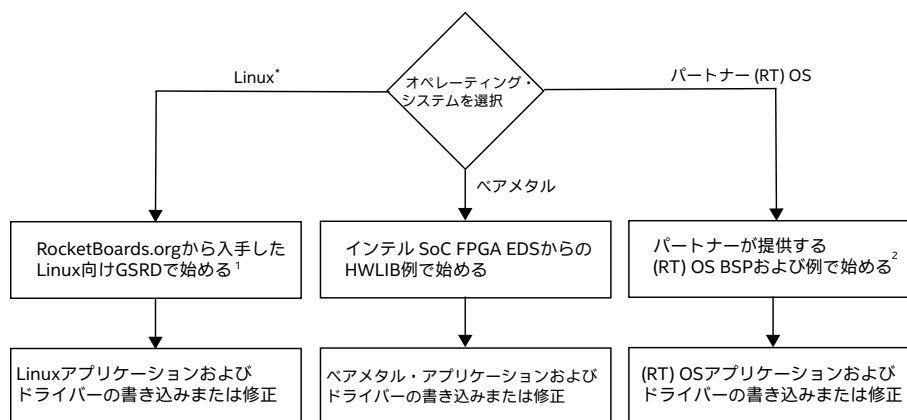
記載されている推奨事項に従い、最終アプリケーションのパフォーマンス、サポート、および市場投入までの時間に関する要件に合うコンポーネントをソフトウェア・プラットフォームに選択する必要があります。

### 5.2. ソフトウェア開発プラットフォームのコンポーネントの組み立て

ソフトウェア開発プラットフォームを正常に構築するため、インテル では、HPS システムの適切なコンフィギュレーションとして知られているベースライン・プロジェクトから始めることを推奨しています。その後、最終アプリケーションに合わせてベースライン・プロジェクトを変更することが可能です。

図 21 (51 ページ) は、ソフトウェア開発プラットフォームのコンポーネントを決定するために推奨される手順を表しています。

図 -21: ソフトウェア開発プラットフォームの組み立て



(1) ゴールデン・システム・リファレンス・デザイン (GSRD) は、さまざまなカスタム・ユーザー・デザインの開始点として使用できる一連の必須ソフトウェアおよびハードウェア・システム・コンポーネントを提供します。詳細は、RocketBoards.orgを参照ください。

注意: RocketBoards.orgは、Linuxアプリケーションのデベロッパー、ファームウェア・エンジニアおよびハードウェア・エンジニアに向けたオープンソース・コミュニティです。

(2) パートナーが提供するボード・サポート・パッケージ (BSP) の一部は、ゴールデン・ハードウェア・リファレンス・デザイン (GHRD) をベースにしています。

注意: GHRDはGSRDの一部です。

フローは、次のステップで構成されます。

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。

1. 次のオペレーティング・システムのいずれかを選択します。
  - ベアメタル
  - Linux\*または パートナーのオペレーティング・システム
  - リアルタイム・オペレーティング・システム
2. アプリケーションとドライバー（どちらかもしくは両方）の書き込みおよび更新（どちらかもしくは両方）を行います。

#### 関連情報

[RocketBoards.org Web Page](#)

GSRD の詳細については、検索フィールドに「GSRD」と入力してください。

## 5.3. ゴールデン・ハードウェア・リファレンス・デザイン (GHRD)

GHRD はゴールデン・システム・リファレンス・デザイン (GSRD) の一部であり、インテル Stratix 10 SoC 開発キット向けの完全な HPS デザインを含む インテル Quartus Prime プロジェクトです。GHRD は、ブートソース、SDRAM メモリーおよび、開発ボード上のその他のペリフェラルへの接続を備えています。

HPS の機能を利用することを選択した場合は常に、インテル Stratix 10 SoC を備えるハードウェア・デザインを使用する必要があります。ハードウェア・デザインの目的は、FPGA 部分、HPS のピン・マルチプレクサーと I/O、および DDRAM を含む SoC をコンフィグレーションすることです。ソフトウェア・プロジェクトはすべて、ハードウェア・デザインに依存します。

GHRD は、Quartus Prime デザイン・スイート (QPDS) のすべてのメジャーリリースにおいて回帰テストが行われています。また、既知のハードウェアの問題に対する最新のバグ修正が含まれています。そのため GHRD は、SoC FPGA ハードウェア・システムの優れたコンフィグレーションとして機能することが知られています。

**ガイドライン: 最新の GHRD を新規 SoC FPGA ハードウェア・プロジェクトのベースラインとして使用します。その後、エンド・アプリケーションの必要性に応じてデザインを変更することが可能です。**

GHRD は以下のいずれかから取得可能です。

- Linux 向け GSRD のウェブページこのウェブページには、最良のコンフィグレーションとして知られる最新バージョンが含まれます。
- SoC EDS インストール・フォルダー <SoC EDS Installation directory> \examples\hardware\s10\_soc\_devkit\_ghrd このフォルダーには、対応する SoC EDS のバージョンでサポートされるバージョンが含まれており、SoC EDS で提供される インテル・ハードウェア・ライブラリー (HWLIB) デザイン例のベースとして使用されています。これは最新のコンフィグレーションではない場合があります。

#### 関連情報

[GSRD for Linux page](#)

## 5.4. アプリケーションのオペレーティング・システムの選択

### 5.4.1. Linux または RTOS の使用

インテル Stratix 10 SoC をサポートするオペレーティング・システムは、Linux OS を含めていくつかあります。



詳細は、OS SoC パートナーのエコシステム・ウェブページを参照ください。

SoC FPGA のオペレーティング・システムの選択には、次のような多くの内容が考慮されます。

- オペレーティング・システムの機能
- ライセンス条項
- オペレーティング・システムに基づく共同ソフトウェア・プロジェクトおよびフレームワークの可用性
- 利用可能なデバイスドライバおよびリファレンス・ソフトウェア
- 社内のレガシーコードおよび、オペレーティング・システムに関する知識
- システムのリアルタイム要件
- アプリケーションに必要な機能上の安全性およびその他の認証

アプリケーションに適したオペレーティング・システムを選択するには、SoC FPGA で利用可能なオープンソースおよび市販のオペレーティング・システムで提供される機能とサポートサービスに関して理解する必要があります。インテルの OS パートナーのウェブサイトには、選択を行う際に参考となる情報が提供されています。

インテル は、Yocto Project と互換性のある Ångström ディストリビューションをサポートします。

パートナー OS プロバイダーは、SoC FPGA デバイスに向けたボード・サポート・パッケージおよび商用サポートを提供しています。Linux コミュニティーもまた、SoC FPGA デバイスに向けたボード・サポート・パッケージおよびコミュニティ・サポートを提供しています。

オペレーティング・システムのリアルタイム・パフォーマンスとベアメタル・アプリケーションに関してはいくつかの誤解があります。Arm Cortex\* の A クラス・プロセッサの場合、ランタイム・アプリケーションを管理するために提供される機能に加え、リアルタイム・オペレーティング・システムはプロセッサのリソースを効率的に使用する機能をいくつか提供します。

これらの効率性はアプリケーションに十分なリアルタイム・パフォーマンスをもたらし、利用可能なデバイスドライバ、ミドルウェア・パッケージ、ソフトウェア・アプリケーションおよびサポートサービスの大部分を継承できるようになります。オペレーティング・システムを選択する際にはこの内容を考慮する必要があります。

### 5.4.2. ベアメタル・アプリケーションの開発

HPS はベアメタル・コンフィグレーション (オペレーティング・システムなし) で使用でき、インテルでは、高レベル API および低レベルマクロで構成されるハードウェア・ライブラリー (HWLIB) をほとんどの HPS ペリフェラルに提供しています。

ベアメタル・ソフトウェアはボードの起動に通常使用されますが、ベアメタルは実際のアプリケーション・プラットフォームとしても使用できます。HPS に向けてベアメタル・アプリケーションを開発するには、ランタイム機能の開発について理解し、ベアメタル・アプリケーションがマイクロプロセッサ・ユニット (MPU) のサブシステムで利用可能なリソースを効率的に使用できるようにする必要があります。

次に例を示します。

- 一般的なベアメタル・アプリケーションは単一のコアのみを使用します。MPU のサブシステムをフル活用するには、4 つのコアすべてを管理するランタイム機能を開発する必要があります。
- アプリケーションが複雑になるにつれ、プロセスの管理とスケジューリング、プロセス間通信の処理、およびアプリケーション内のイベントの同期を行う機能を構築することが必要になる場合があります。

そのため、小さな Lightweight RTOS でさえも、MPU サブシステムのリソースをより効率的に使用する簡単なスケジューリング、プロセス間通信、および割り込み処理機能を提供します。

### 5.4.3. ベアメタル・フレームワークとしてのブートローダーの使用

アプリケーションが比較的シンプルで、マルチコアやマルチタスクなどの複雑な機能を必要としない場合、アプリケーションをブートローダーに含めることが 1 つの選択肢になります。

アプリケーションをブートローダーに含めることには次の利点があります。

- ブート時間短縮の可能性
- 大容量ストレージやネットワークなど、ブートローダーにすでに実装されている機能へのアクセス

以下は、利用可能なブートローダーとそのソースコードです。

- U-Boot—オープンソース GPL ライセンス
- UEFI—オープンソース BSD ライセンス
- Arm Trusted Firmware (ATF)—オープンソース BSD ライセンス

### 5.4.4. 対称型および非対称型マルチプロセッシング・モード (SMP および AMP)

インテル Stratix 10 HPS のクアドコア Arm Cortex-A53 MPCore\* は、対称型マルチプロセッシング (SMP) と非対称型マルチプロセッシング (AMP) のオペレーティング・モードをサポートします。

SMP モードでは、単一のオペレーティング・システムのインスタンスが 4 つのコアすべてを制御します。SMP コンフィグレーションは、さまざまなオペレーティング・システム・メーカーによってサポートされており、マルチプロセッシングの最も一般的で複雑ではないコンフィグレーション・モードです。

Linux および商業用に開発されたオペレーティング・システムは、CPU コアのリソースを最大限に活用する機能を提供しており、それらを効率的に使用することで、最適なパフォーマンスと使いやすさを実現します。例えば、SMP 対応のオペレーティング・システムは、プロセッサ親和性を設定するオプションを提供します。つまり、各タスクまたはスレッドを特定のコアで実行するように割り当てることができます。この機能により、それぞれの Arm Cortex-A53 コアへのワークロード分散をより適切に制御し、AMP の代替としてシステムの応答性を高めることができます。

**ガイドライン: 市販のオペレーティング・システムで利用可能なパフォーマンスおよび最適化を理解し、SMP 対応のオペレーティング・システムまたは RTOS が、要求するパフォーマンスおよびリアルタイム要件を満たしているかを確認します。**

AMP のコンフィグレーションでは、最大 4 つの異なるオペレーティング・システムを 4 つの Cortex-A53 コアで動作させることができるため、より有効な組み合わせの実現が可能になります。AMP と SMP を組み合わせ、2 つのコアで SMP を実行し、残りの 2 つのコアで AMP を実行することも可能です。

#### 特別に考慮する内容

- AMP は、プロセスの管理とスケジューリング、プロセス間通信の処理、イベントの同期、およびオペレーティング・システムの 2 つのインスタンス間の安全なプロセス管理の手法を十分に理解している場合のみ使用します。
- OS プロバイダーは一般的に、提供するオペレーティング・システムを AMP モードで使用するためのサポートを提供しません。そのため、それらを AMP モードで使用する場合は通常、特別なサポート契約が必要になります。
- AMP を使用する場合は、Cortex-A53 のバーチャライゼーション・フィーチャーを利用することが推奨されます。Cortex-A53 にはバーチャライゼーションに向けたネイティブ・ハードウェア・サポートが含まれており、AMP のリソース共有問題のほとんどを解決します。

## 5.5. Linux に向けたソフトウェア開発プラットフォームの構築

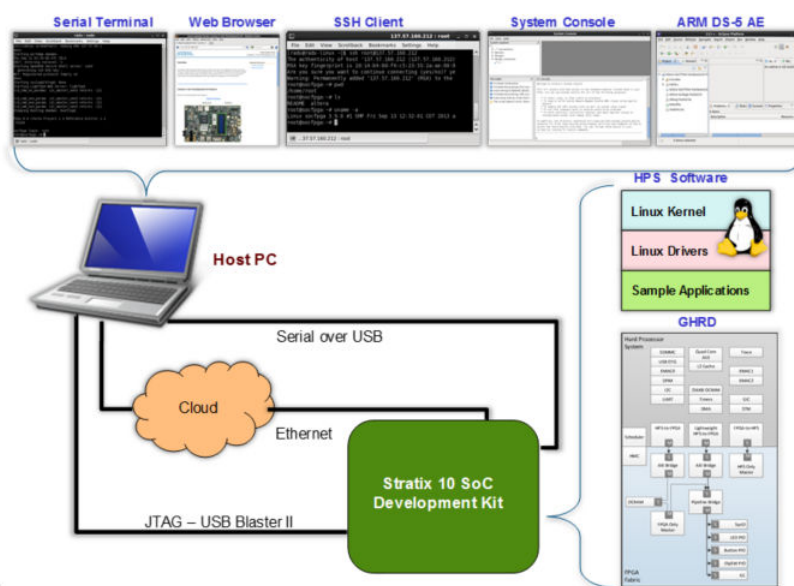
この章では、エンド・アプリケーションのオペレーティング・システムに Linux を選択した場合のデザイン・ガイドラインを示します。

### 5.5.1. Linux 向けゴールデン・システム・リファレンス・デザイン (GSRD)

インテル では、以下で構成される Linux 向けの GSRD を提供しています。

- GHRD - QPDS プロジェクト
- U-Boot ベースのリファレンス・ブートローダー
- リファレンス Linux BSP
- サンプル Linux アプリケーション

図 -22: Linux 向け GSRD の概要



Linux 向けの GSRD は十分にテストされた優れたデザインとして知られ、HPS と FPGA リソースをどちらも使用するシステムを紹介しています。これは、ベースライン・プロジェクトとして使用することを目的にしています。

**ガイドライン: Linux ソフトウェア開発プラットフォームを正常に構築するため、インテルでは、最新の GSRD をベースライン・プロジェクトとして使用することを推奨しています。**

GSRD は、インテル SoC 開発ボードをターゲットとしており、ソースおよびプリコンパイルの両方の形式で提供されます。Rocketboards.org から GSRD をダウンロードし、ご自身のアプリケーションの必要性に応じて変更します。

#### 関連情報

[RocketBoards.org Web Page](#)

GSRD の詳細については、検索フィールドに「GSRD」と入力してください。

### 5.5.2. ソースコード管理における考慮事項

GSRD のビルドプロセスは、オンラインで入手可能ないくつかの Git ツリーに依存しています。以下にその一部を示します。

表 11. Git ツリーのリンク

Git ツリー	リンク
インテル SoC FPGA Linux カーネル	<a href="https://github.com/altera-opensource/linux-socfpga">https://github.com/altera-opensource/linux-socfpga</a>
インテル SoC FPGA Linux デザイン	<a href="https://github.com/altera-opensource/linux-refdesigns">https://github.com/altera-opensource/linux-refdesigns</a>
インテル SoC FPGA Angstrom レシピ	<a href="https://github.com/altera-opensource/angstrom-socfpga">https://github.com/altera-opensource/angstrom-socfpga</a>

**注意:** インテルは、Linux の有効化、メインラインへのアップストリームおよび Linux コミュニティーとのコラボレーションを行っています。インテルでは、最新の安定版カーネル (N) および最新の LTSI カーネル (M) の 2 つのカーネルバージョンを提供し、以前の Linux カーネルバージョン (N-1、M-1) のサポートを終了します。N、N-1、M、M-1 バージョンはいつでもカーネルのリポジトリから入手可能です。古いカーネルのバージョンは削除されます。

**ガイドライン:** インテル・オープンソース・サイトにあるリポジトリのコンテンツが永続的に利用可能であると想定せず、独自に Git リポジトリを管理してください。Git リポジトリの管理は、Git サービス・プロバイダーを利用するなどさまざまな方法で実現できます。独自の Git リポジトリを管理する利点には、ビルドの再現性、ソースコードの管理、Git によって有効になった分散モデルの活用などがあります。

GSRD は、Yocto レシピを使用し構築された Angstrom の rootfilesystem を使用します。レシピはさまざまなオープンソースのパッケージ・ソースを取り込み、それらを rootfilesystem に構築しています。これらのレシピの一部は汎用であり、特定のバージョンを参照しないため、最終結果はビルドごとに異なる場合があります。

**ガイドライン:** Angstrom の rootfilesystem を再構築し、再現性が必要な場合は、ビルドに使用した Yocto ダウンロード・フォルダーのコピーを保持しておく必要があります。

### 5.6. ベアメタル・アプリケーションに向けたソフトウェア開発プラットフォームの構築

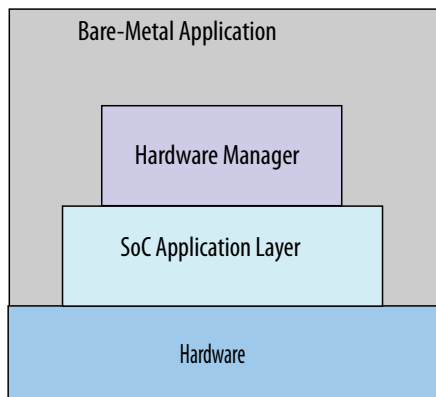
HWLIB は、SoC EDS で提供される一連の低レベル・ベアメタル・ソフトウェア・ユーティリティであり、HPS のさまざまなコンポーネントを制御するためにデザインされています。HWLIB はまた、インテルの OS パートナーにおけるオペレーティング・システムに向けたボード・サポート・パッケージの構築において一般的に使用されています。

HWLIB には 2 つのコンポーネントがあります。

- SoC 抽象化レイヤー (SoCAL): HPS アドレス空間内のデバイスレジスターの直接アクセスと制御を可能にするレジスター抽象化レイヤーです。
- ハードウェア・マネージャー (HWMgr): より高度なユース・ケース・シナリオに向けてより複雑な機能とドライバを提供する API です。



図 -23: HWLIB の概要



注意: SoCAL および HWMgr は、すべてのハードウェアを網羅しているわけではありません。そのため、アプリケーションによってはカスタムコードの記述が必要になる場合があります。

HWLib を使用するソフトウェア・アプリケーションには、MPU サブシステムのリソースを管理するためのランタイム・プロビジョニングが必要です。これらのプロビジョニングは通常、オペレーティング・システムが提供します。

ガイドライン: アプリケーションを管理するランタイム・プロビジョニングの開発について十分に理解している場合にのみ、HWLIB を使用します。

ガイドライン: <SoC EDS installation folder>/embedded/examples/software/ で提供される HWLIB 例を、ベアメタル開発の開始点として使用します。

HWLIB についての詳細は、以下のドキュメントを参照ください。

- *Intel SoC FPGA Embedded Design Suite User Guide*
- インテル Wiki の Getting Started with HWLIBs Bare-metal Development ウェブページ

#### 関連情報

- [Intel SoC FPGA Embedded Design Suite](#)
- [SoCEDSGettingStarted](#)  
この Wiki ページには、インテル SoC FPGA エンベデッド開発スイート (SoC EDS) を開始する際に役立つスタートガイドのリストが含まれています。

## 5.7. パートナー OS または RTOS に向けたソフトウェア開発プラットフォームの構築

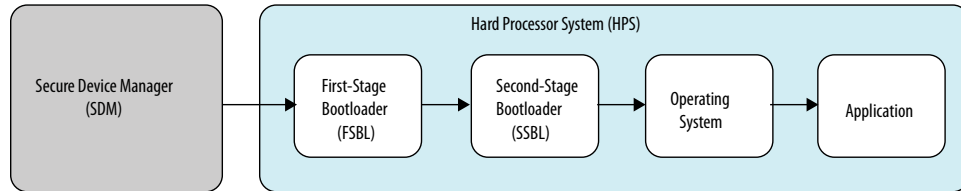
パートナー OS のプロバイダーは、インテル SoC FPGA デバイスに向けたサポートパッケージおよび商用サポートを提供します。サポートには通常、プロジェクトを開始するための例と関連するドキュメントが含まれています。

パートナー OS または RTOS をターゲットとするソフトウェア開発プラットフォームの構築方法に関する詳細は、パートナーが提供するドキュメントおよびサポートサービスを参照してください。

## 5.8. ブートローダー・ソフトウェアの選択

Stratix 10 SoC HPS の一般的なブートフローを次の図に表しています。

図 -24: 一般的な インテル Stratix 10 SoC のブートフロー



ブートローダー・ソフトウェアは、ソフトウェア開発プラットフォームにおいて最も重要なコンポーネントの1つです。ブートローダーはシステムを初期化し、次のブートイメージであるオペレーティング・システムまたはベアメタル・アプリケーションに制御をロードし渡します。

インテル Stratix 10 SoC ブートローダー・ソフトウェアは2つの異なる段階に分かれています。

- 第1ステージ・ブートローダー (FSBL) - FPGA コンフィグレーション・ビットストリームから HPS 側のオンチップメモリに SDM によってロードされます。
  - HPS をコンフィグレーションするために不可欠な初期のハードウェア設定を提供します
  - HPS のフラッシュおよびペリフェラル・コンポーネントを制御するソフトウェア機能
  - 早期デバッグおよびトラブルシューティングを可能にするユーティリティ
- 第2ステージ・ブートローダー (SSBL) - FSBL によって DDRAM にロードされ、ネットワーク・アクセス、コマンドライン・インターフェイス、スクリプティング・サポートなど、潜在的に FSBL よりもはるかに多くの機能を備えます。

インテル では、いくつかのブートローダーのオプションを提供しています。

- **U-Boot ブートローダー:** オープンソース・コミュニティから入手可能な機能の一部を継承しており、Linux OS ユーザーの間で広く利用されています。U-Boot ブートローダーは GPL ライセンスによって管理されています。SoC EDS の一部として入手可能で、Linux 向け GSRD に使用されます。
- **UEFI ブートローダー:** 機能が豊富で、RTOS ユーザーに広く利用されています。オープンソースの BSD スタイルライセンスで管理されています。
- **ATF (ARM Trusted Firmware) ブートローダー:** UEFI で利用され、第1ステージ・ブートローダーのみを提供します。BSD スタイルライセンスを使用し、SSBL に代わりベアメタル・アプリケーションを直接ロードするために使用可能です。

**ガイドライン:** ソフトウェア開発プラットフォームに適したブートローダーを選択するには、最新バージョンを使用し、また、GPL およびオープンソースの BSD ライセンスについて十分に理解し、どのライセンス条項がご自身の要求に最適かを検討します。

一般的な HPS システムには数百のレジスターがあり、それらを MPU サブシステム、ネットワークオンチップ・インターコネクト・コンポーネント、DDRAM メモリー、フラッシュ・ブート・ソース、およびペリフェラル・インターフェイスの特定のコンフィグレーションに設定する必要があります。

**ガイドライン:** 必要とされる初期化設定の量を考えると、ブートローダーをゼロから作成することは推奨されません。提供されているブートローダーのオプションには、HPS のさまざまな部分に向けた最適なデフォルト・コンフィグレーション設定が含まれています。

## 5.9. 開発、デバッグ、およびトレースに向けたソフトウェア・ツールの選択

この章では、さまざまなソフトウェア開発ツールを選択するためのデザイン上の考慮事項について説明します。



**注意:** 特定のパートナー OS または RTOS を使用する場合に必要な特定のツールについては、その OS ベンダーに相談、または提供されているドキュメントを参照してください。一部の OS ベンダーは、そのオペレーティング・システムで使用することが推奨されるツールのフルセットも提供しています。

**注意:** 開発、コンパイルおよびデバッグに利用可能なツールに関して十分に理解してください。サポートされているツールのリストは、インテル FPGA エコシステムのウェブページにあります。

#### 関連情報

[インテル FPGA エコシステムのウェブページ](#)

### 5.9.1. ソフトウェア・ビルド・ツールの選択

使用するソフトウェア開発ツールとそのバージョンを選択する必要があります。

- コンパイラー
- アセンブラー
- リンカー
- アーカイバー

Arm Development Studio 5\* インテル SoC FPGA エディションには、Arm V8-A ソフトウェア・ビルド・ツールである Arm ヘアメタル・コンパイラー 6 が含まれます。

U-Boot は、Linaro\*リリース・ウェブページで入手可能な Linaro GCC コンパイラーでコンパイルされます。

その他の開発ツールも、他のサードパーティー・プロバイダーから提供されています。

#### 関連情報

[Linaro Releases](#)

### 5.9.2. ソフトウェア・デバッグ・ツールの選択

使用するソフトウェア・デバッグ・ツールを決定する必要があります。

Arm DS-5\* インテル SoC FPGA Edition には、フル機能を備えた Eclipse ベースのデバッグ環境が含まれます。そのほかには Lauterbach\* T32 など、サードパーティー・プロバイダーから提供されているデバッグツールもあります。

デバッグツールには、インテル SoC FPGA デバイスへの JTAG 接続が必要です。JTAG 接続は、以下を使用し実現可能です。

- インテル Stratix 10 SoC 開発キットなどで入手可能なエンベデッド・Intel® FPGA Download Cable II
- 外部 JTAG ハードウェア (Lauterbach T32 ツール使用時に場合によっては必要になるものと同様のもの)

### 5.9.3. ソフトウェア・トレース・ツールの選択

トレースは、パフォーマンスのボトルネックのプロファイリング、クラッシュシナリオのデバッグ、および複雑なケースのデバッグに非常に有効です。トレースの実行には 2 つの方法があります。

- **非リアルタイム**: トレースデータをシステムメモリ (SDRAM など) または組み込みトレースバッファに格納後、システムを停止し、JTAG を介してトレース情報をダウンロードし分析します。
- **リアルタイム**: 外部アダプターを使用し、トレースポートからトレースデータをキャプチャーします。ターゲットボードはこのシナリオをサポートしている必要があります。

デバッグツールは通常、エンベデッド・ソフトウェア・プログラムの実行におけるトレースも提供しますが、外部ハードウェアが必要になる場合があります。例えば、SoC EDS で提供される Arm DS-5 インテル SoC FPGA Edition は、非リアルタイムおよびリアルタイムのトレースをどちらもサポートします。リアルタイム・トレースに使用する場合は、Arm DSTREAM と呼ばれる外部トレースユニットが必要です。

Lauterbach T32 もまた、リアルタイム・トレースに外部ハードウェアが必要です。

### 5.10. ブートおよびコンフィグレーションにおける考慮事項

インテル Stratix 10 SoC HPS はブート ROM を備えていません。代わりに SDM には、初期の FPGA コンフィグレーション・ビットストリームをロードする BootROM があります。このビットストリームには、HPS の第 1 ステージ・ブートローダー (FSBL) バイナリーも含まれています。

詳細は、*Intel Stratix 10 SoC FPGA Boot User Guide* を参照ください。

#### 関連情報

[Intel Stratix 10 SoC FPGA Boot User Guide](#)

#### 5.10.1. コンフィグレーション・ソース

初期の FPGA コンフィグレーションおよび HPS FSBL は初期コンフィグレーション・ビットストリームの一部であり、いくつかのソースから取得可能です。

- **Avalon -St データソース**: Avalon -ST マスターはビットストリームを提供します。
- **JTAG インターフェイス**: 外部 JTAG マスター (通常はホストツールで駆動されます) はビットストリームを提供します。
- **SDM フラッシュ**: SDM 側に接続されたフラッシュデバイスはビットストリームを提供します。

次のタイプのフラッシュデバイスを SDM に接続できます。

表 12. フラッシュのタイプ別サポート状況

フラッシュタイプ	サポート状況
QSPI	現在 インテル Quartus Prime プロ・エディション 18.1 リリースでサポートされています。
SD/eMMC	今後の インテル Quartus Prime プロ・エディションのリリースにてサポートされます。

#### 5.10.2. コンフィグレーション・フラッシュ

次の QSPI デバイスが インテル Stratix 10 SoC コンフィグレーションに検証されています。



表 13. QSPI デバイス

ベンダー	部品番号	容量
Micron*	MT25QU128	128 Mb
Micron	MT25QU256	256 Mb
Micron	MT25QU512	512 Mb
Micron	MT25QU01G	1 Gb
Micron	MT25QU02G	2 Gb
Macronix*	MX25U128	128 Mb
Macronix	MX25U256	256 Mb
Macronix	MX25U512	512 Mb
Macronix	MX66U512	512 Mb
Macronix	MX66U1G	1 Gb
Macronix	MX66U2G	2 Gb

ガイドライン: FPGA をフラッシュからコンフィグレーションする場合、互換性のある QSPI デバイスを選択します。

ガイドライン: デザインに合う QSPI デバイスを選択します。より大きなデバイスを使用することで、デザインのビットストリーム・サイズの増加に備えます。

### 5.10.3. コンフィグレーション・クロック

ガイドライン: インテル Quartus Prime プロ・エディション GUI で、選択した QSPI フラッシュデバイスの能力に合うコンフィグレーション・クロック速度を選択します。

### 5.10.4. HPS ブートオプションの選択

インテル Quartus Prime プロ・エディションの「FPGA Interfaces」タブにある「HPS Boot Source」サブウィンドウから、コンフィグレーションとブートモードを選択する必要があります。

- **FPGA Configuration First:** SDM は、FPGA コアおよびすべてのペリフェラル I/O をコンフィグレーションした後に、FSBL を HPS オンチップ RAM にロードし、HPS をリセットから解除します。初期コンフィグレーション中にエラーが発生した場合、HPS はリセットから解除されません。
- **HPS First:** SDM は HPS SDRAM に必要な I/O のみをコンフィグレーションし、HPS をリセットから解除する前に FSBL を HPS オンチップ RAM にロードします。FPGA コアおよびほかの未使用の I/O はコンフィグレーションされません。HPS が FPGA の残りの部分をコンフィグレーションします。

### 5.10.5. HPS ブートソース

HPS FSBL は、初期の FPGA コンフィグレーション・ビットストリームに含まれます。HPS SSBL はいくつかの位置に配置できます。

表 14. HPS SSBL のサポート状況

HPS SSBL の位置	サポート状況
SDM QSPI	現在 インテル Quartus Prime プロ・エディション 18.1 リリースでサポートされています。
HPS SD/eMMC	現在 インテル Quartus Prime プロ・エディション 18.1 リリースでサポートされています。
HPS NAND	今後の インテル Quartus Prime プロ・エディション・リリースにてサポートされます。

**ガイドライン:** インテル では、HPS SSBL を HPS SD/eMMC フラッシュに配置することを推奨しています。

### 5.10.6. リモート・システム・アップデート (RSU)

インテル Stratix 10 SoC は、リモート・システム・アップデート (RSU) の機能をサポートしています。この機能を使用する場合、複数のプロダクション・イメージをフェイルセーフのファクトリー・イメージとともに外部 SDM フラッシュに保存するオプションがあります。POR を終了すると、SDM は特定のシーケンスでプロダクション・イメージのロードを試みます。すべてのプロダクション・イメージのロードに失敗した場合、フェイルセーフのファクトリー・イメージがロードされます。

RSU の機能に関する詳細は、*Intel Stratix 10 SoC Remote System Update (RSU) User Guide* を参照ください。

#### 関連情報

[Intel Stratix 10 SoC Remote System Update \(RSU\) User Guide](#)

### 5.11. システムリセットにおける考慮事項

4 つのウォッチドッグ・タイマーのいずれかが設定時間に達し、SDM へのシステムリセット要求が生成されると、SDM は次の 3 つのタイプのシステムリセットのいずれかを実行します。

- HPS Cold reset
- HPS Warm reset
- HPS Cold reset and trigger remote update

**注意:** インテル Quartus Prime プロ・エディション・ツール内から、これらの 3 つのオプションの 1 つを選択できます。

インテル Quartus Prime プロ・エディション・ツールで、「HPS Clocks and resets」タブ、そして「Resets」タブを選択し、「Enable watchdog reset」のチェックボックスをクリックします。その後、「How SDM handles HPS watchdog reset」ラベルのプルダウンメニューにある 3 つのオプションから 1 つを選択します。



- **HPS Cold reset**
  - **HPS への影響**—SDM はプロセッサをリセット状態に維持します。SDM は、コールドリセットの前にデバイスにロードされたものと同じビットストリームから、FSBL を HPS オンチップメモリにロードします。正常に終了すると、SDM は HPS のリセットを解除し、プロセッサはリセット例外アドレスからコードの実行を開始します。
  - **FPGA への影響**—リセット中に FPGA コア・ファブリックは変更されません。リセット終了後、ソフトウェアは FPGA 部分をリコンフィグレーションするかどうかを決定します。
- **HPS Warm reset**
  - **HPS への影響**—SDM はプロセッサをリセット状態に維持します。FSBL はウォームリセット中にオンチップ RAM に維持されます。SDM がプロセッサのリセットを解除し、プロセッサはオンチップ RAM の FSBL を実行します。
  - **FPGA への影響**—FPGA 部分はリセット中に維持されます。リセット終了後、ソフトウェアは FPGA 部分をリコンフィグレーションするかどうかを決定します。
- **HPS Cold reset and trigger a remote Update**
  - **HPS への影響**—SDM はプロセッサをリセット状態に維持します。SDM は、次に有効な \*.pof イメージまたはファクトリー・イメージから FSBL を HPS オンチップメモリにロードします。\*.pof には、SoC の FPGA 部分をコンフィグレーションするためのデータと FSBL ペイロードが含まれます。正常に終了すると、SDM は HPS のリセットを解除し、プロセッサがリセット例外アドレスからコードの実行を開始します。
  - **FPGA への影響**—FPGA 部分がまず消去された後に、次に有効なコア RBF またはファクトリー・コア RBF でリコンフィグレーションされます。有効なファクトリー RBF が常に存在している必要があります。

## 5.12. フラッシュの考慮事項

### 5.12.1. フラッシュのプログラミング方法

SDM に接続されるフラッシュは、インテル Quartus Prime プロ・エディションおよび インテル SoC FPGA エンベデッド開発スイート (SoC EDS) の一部である インテル Quartus Prime プログラマー・ツールでプログラムされます。

**ガイドライン: インテル Quartus Prime プロ・エディションのプログラマーを使用し、SDM フラッシュに書き込みます。**

HPS に接続されるフラッシュのプログラムはご自身で行ってください。いくつかのオプションが利用可能です。

- バススイッチを使用し、プログラミングを行う外部マスターにフラッシュ信号をルーティングする。
- HPS で実行されているソフトウェアを使用しプログラミングを行う。例えば、U-Boot は Arm デバッガーまたはシステムコンソールでロードし、フラッシュのプログラムに使用できます。

**ガイドライン:** ボードデザインに影響を与える場合や、ツールサポートが追加で必要になる場合があるため、HPS フラッシュのプログラミング方法はプロジェクト・ライフサイクルの早期段階において計画します。

### 5.12.2. FPGA コンフィグレーションおよび HPS 大容量ストレージへのシングルフラッシュの使用

SDM に接続される QSPI デバイスには、HPS から直接アクセスすることも可能です。ただし、この方法は速度を大幅に低下させます。最終アプリケーションにおいて速度低下が許容されるかどうかを判断してください。

参考のため、パフォーマンスの数値を以下に示します。

- HPS eMMC の最大読み取り速度: 50Mbytes/秒
- HPS SD の最大読み取り速度: 25Mbytes/秒
- SDM QSPI からの HPS の最大読み取り速度: 4Mbytes/秒

**ガイドライン:** 最良のパフォーマンスを実現するため、インテル では、HPS に接続されるフラッシュデバイスを HPS の大容量ストレージに使用することを推奨しています。

### 5.13. エンベデッド・ソフトウェアのデバッグとトレース

HPS デバッグ・アクセス・ポート (DAP) には、JTAG としてコンフィグレーションされる専用 HPS ピン、もしくは FPGA JTAG インターフェイス・ピンを介してアクセスすることができます。

FPGA JTAG ピンを介して HPS JTAG インターフェイスにアクセスするオプションは、インテル Quartus Prime プロ・エディションのプロジェクトで利用可能です。

電源投入時に、FPGA は JTAG チェーンの最初のデバイスとして表示されます。HPS JTAG インターフェイスを FPGA JTAG ピンで使えるようにするイメージで FPGA がコンフィグレーションされると、HPS は JTAG チェーンの最初のインターフェイスとして表示され、FPGA は 2 番目のインターフェイスとして表示されます。そのため、インテル Quartus Prime プロ・エディション・プログラマーなどの FPGA ツールは、電源投入時に使用する場合と FPGA コンフィグレーション後に使用する場合は異なる接続設定を必要とします。

**ガイドライン:** 開発およびフィールド問題のデバッグと診断に使用できる、ボードへの利用可能な JTAG 接続が必要です。

HPS は、2 つのトレース・インターフェイスを HPS 専用 I/O もしくは FPGA I/O のいずれかで提供します。HPS 専用 I/O で提供されるインターフェイスは低速トレース・インターフェイスであり、低帯域幅トラック (低周波数で動作する MPU など) をトレースするために使用できます。

トレース帯域幅を向上させるために、標準トレース・インターフェイスである FPGA への 32 ビット・シングル・データ・レート・インターフェイスを使用することができます。

トレースベンダーより提供されているデータシートを参照し、トレースバスに終端が必要かの判断を行ってください。トレースベンダーが必要と示す終端を含めない場合、トレースデータの破損につながる、またはインターフェイスの最大動作周波数が制限される可能性があります。





## 5.14. インテル Stratix 10 SoC FPGA 向けエンベデッド・ソフトウェアのデザイン・ガイドライン改訂履歴

表 15. Stratix 10 SoC FPGA 向けエンベデッド・ソフトウェアのデザイン・ガイドライン改訂履歴

ドキュメント・バージョン	変更内容
2019.04.17	メンテナンス・リリース
2019.03.19	リモート・システム・アップデート (RSU) の内容を追加しました。
2018.12.24	<i>Intel Stratix 10 SoC FPGA Boot User Guide</i> へのリンクを追加しました。
2018.09.24	新しく「システムリセットにおける考慮事項」の章を追加しました。この章では、SDM が実行する 3 つの種類のリセットを説明しています。
2018.05.07	次の内容を追加しました。 <ul style="list-style-type: none"><li>• 次のシステムに対するソフトウェア開発プラットフォームの構築<ul style="list-style-type: none"><li>– Linux</li><li>– ベアメタル・アプリケーション</li><li>– パートナー OS または RTOS</li></ul></li><li>• ブートローダー・ソフトウェアの選択</li><li>• 開発、デバッグ、およびトレースに向けたソフトウェア・ツールの選択</li><li>• ブートおよびコンフィグレーションにおける考慮事項</li><li>• フラッシュの考慮事項</li><li>• エンベデッド・ソフトウェアのデバッグとトレース</li></ul>
2018.03.01	メンテナンス・リリース
2017.11.06	初版

## 6. Stratix 10 SoC FPGA に関連する推奨リソース

---

### 6.1. デバイスのドキュメント

- [Intel Stratix 10 Hard Processor System Technical Reference Manual](#)
- [インテル Stratix 10 デバイス・データシート](#)
- [インテル FPGA デバイス用ピンアウトファイル](#)
- [Hard Processor System Pin Information for Intel Stratix 10 Devices \(version 2017.10.09\)](#)
- [インテル Stratix 10 デバイスファミリー・ピン接続ガイドライン](#)
- [インテル Stratix 10 デバイスのデザイン・ガイドライン](#)
- [インテル Stratix 10 高性能デザイン・ハンドブック](#)
- [External Memory Interfaces Intel Stratix 10 FPGA IP User Guide](#)
- [Intel Stratix 10 External Memory Interface Pin Information for Devices \(version 2018.1.23\)](#)
- [AN 692: インテル Cyclone® 10 GX、インテル Arria® 10、および インテル Stratix 10 デバイスの電源シーケンスについての考慮事項](#)
- [Differences Among Intel SoC Device Families](#)
- [Intel Stratix 10 SoC FPGA Boot User Guide](#)
- [Early Power Estimator for Intel Stratix 10 FPGAs User Guide](#)

以下については、お住まいの地域のインテル FPGA サポート担当にご連絡ください。

- [インテル Stratix 10 SX ES Device Errata](#)

### 6.2. ツールとソフトウェアのウェブページ

- [Arm DS-5 インテル SoC FPGA Edition ツールキット](#)
- [インテル SoC FPGA エンベデッド開発スイート \(SoC EDS\)](#)
- [インテル Quartus Prime 開発ソフトウェア・スイート](#)
- [Mentor Graphics\\* Embedded for Intel](#)
- [External Memory Interface \(EMIF\) Spec Estimator](#)
- [RocketBoards.org - SoC FPGA 上の Linux に向けて](#)



### 6.3. インテル Stratix 10 SoC FPGA に関連する推奨リソースの改訂履歴

表 16. Stratix 10 SoC FPGA に関連する推奨リソースの改訂履歴

ドキュメント・バージョン	変更内容
2019.04.17	メンテナンス・リリース
2019.03.19	メンテナンス・リリース
2018.12.24	<i>Intel Stratix 10 SoC FPGA Boot User Guide</i> へのリンクを追加しました。
2018.09.24	メンテナンス・リリース
2018.05.07	メンテナンス・リリース
2018.03.01	メンテナンス・リリース
2017.11.06	初版