

はじめに

LogiCORE™ の AXI インターコネクト IP は、1 つ以上のメモリマップ方式 AXI マスター デバイスと、1 つ以上のメモリマップ方式のスレーブ デバイスを接続します。AXI インターフェイスは、ARM® の AMBA® AXI 仕様のバージョン 4 に準拠しています。この仕様には AXI4-Lite 制御レジスタ インターフェイスのサブセットも含まれます。

注記：AXI インターコネクト コアは、メモリマップ方式の転送のみを前提に構成されているため、AXI4-Stream による転送には適用できません。AXI4-Stream インターフェイスを備えた IP は、通常同種の IP どうし、および DMA IP と接続されます。

AXI インターコネクト コアは、Xilinx® Platform Studio (XPS) ソフトウェアに含まれる、非暗号化、ライセンス不要 (無償) のプロセッサ コア (pcore) として提供されます。これは、ISE® Design Suite でも提供され、CORE Generator ツールフローを介してエンベデッド以外のデザインで使用できます。

機能

XPS ツールフローでは、AXI インターコネクト コアの全機能を使用できます。CORE Generator ツールフローでは、このセクションで後述されているように、全機能のサブセットがサポートされています。

XPS でサポートされる機能

XPS ツールフローでサポートされる AXI インターコネクト コアの機能は、次のとおりです。

- AXI プロトコル (AXI3、AXI4、AXI4-Lite) に準拠した次の機能
 - 最大バースト長 256 のインクリメンタル (INCR) バーストが可能です。
 - 宛先が AXI3 スレーブ デバイスの場合、16 ビットを超える AXI4 バーストにはトランザクションを分割する変換を加えます。
 - 複数のアドレス デコード範囲を持つスレーブ デバイスで使用する場合、REGION 出力を生成します。
 - USER 信号が存在する場合、それを各チャネルに伝搬します。その場合、USER 信号幅はチャネルごとに独立して設定できます (オプション)。
 - サービス品質 (QoS) 信号が存在する場合、それを伝搬します。この信号は AXI インターコネクト コアでは使用しません (オプション)。
- インターフェイス データ幅
 - AXI4 : 32、64、128、256、512、1024 ビット
 - AXI4-Lite : 32 ビット
- アドレス幅 32 ビット

LogiCORE IP					
コア仕様					
サポート対象のデバイスファミリ (1)	Zyng™-7000、Artix™-7、Virtex®-7、Kintex™-7、Virtex-6、Spartan®-6				
サポート対象のユーザー インターフェイス	AXI4、AXI4-Lite、AXI3				
	リソース				周波数
コンフィギュレーション	LUT	FF	DSP スライス	ブロック RAM	最大周波数
Config1	N/A	N/A	N/A	N/A	N/A
コア付属品					
資料	Product 製品仕様				
デザイン ファイル	Verilog、VHDL				
デザインの例	8 ページの図 1				
テストベンチ	なし				
制約ファイル	ユーザー制約ファイル (UCF)				
シミュレーションモデル	なし				
テスト済みデザイン ツール					
デザイン入力ツール	ISE Design Suite 13.4 PlanAhead™ ツール、XPS				
シミュレーション (2)	Mentor Graphics ModelSim、 Cadence Incisive Enterprise Simulator (IES)				
合成ツール	XST 13.4				
サポート					
ザイリンクスのウェブサイト japan.xilinx.com/support より提供					

1. サポートされる派生デバイスのリストは、[IDS Embedded Edition の派生デバイス サポート](#)を参照してください。
2. ツールのサポート対象バージョンは、『[ISE Design Suite 13 : リリース ノート ガイド](#)』を参照してください。

XPS でサポートされる機能 (続き)

- コアのスレーブ インターフェイス (SI) は、1 ～ 16 SI スロットを含むように構成でき、接続された最大 16 個のマスター デバイスからのトランザクションを受け入れます。マスター インターフェイス (MI) は、1 ～ 16 MI スロットを含むように構成でき、接続された最大 16 個のスレーブ デバイスへのトランザクションを発行します。
- マスターとスレーブを 1 対 1 で接続する場合、AXI インターコネクト コアでは必要に応じてアドレス範囲チェックを実行できます。また、データ幅、クロック レート、プロトコルの変換、レジスタのパイプライン処理、データパスのバッファ処理などのオプション機能も実行可能です。
- マスターとスレーブを 1 対 1 で接続し、変換またはアドレス範囲チェックのいずれも実行しない場合、AXI インターコネクト コアはワイヤとしてインプリメントされ、リソースを使用せず、遅延やレイテンシも発生しません。
- ビルトイン データ幅変換
 - マスターおよびスレーブの各接続では、それぞれ独立したデータ幅 (32、64、128、256、512、1024 ビット幅のいずれか) を使用できます。
 - 内部クロスバーは、ネイティブ データ幅が 32、64、128、256、512、1024 ビットになるようなコンフィギュレーションが可能です。
 - クロスバーのネイティブ データ幅と異なる幅のマスターおよびスレーブ接続ではデータ幅を変換します。
 - よりデータ幅の広いインターフェイスに変換する場合 (アップサイジング)、データをパック (結合) します。それには、アドレス チャネル制御信号によって許可する (CACHE 変更許可ビットをアサートする) 必要があります。
 - よりデータ幅の狭いインターフェイスに変換 (ダウンサイジング) する際に、そのままでは最大バースト長を超えてしまう場合は、バースト トランザクションを複数のトランザクションに分割できます。
- ビルトイン クロック レート変換
 - マスターおよびスレーブの接続は、それぞれ独立したクロック レートを使用できます。
 - 内部クロスバーのネイティブ クロック レートへ整数比 (N:1 および 1:N) の同期変換が可能です。
 - 非同期クロック変換にも対応しています (ただし、同期変換よりも多くのストレージを使用し、より長いレイテンシが発生)。
 - AXI インターコネクト コアは、SI および MI の各スロットに対応するクロック入力に再同期されたリセット信号をエクスポートします。
- ビルトイン AXI4-Lite プロトコル変換
 - AXI インターコネクト コアは、あらゆる比率の AXI4 と AXI4-Lite マスター/スレーブの組み合わせに対する接続が可能です。
 - AXI インターコネクト コアを AXI4-Lite のスレーブに接続した場合、トランザクション ID が保存され、応答転送時にリストアされます。
 - AXI4-Lite のスレーブの場合は ID をサンプリングしたり保存したりする必要はありません。
 - AXI インターコネクト コアは、複数ワードのバーストが実行されるトランザクションなど、AXI4 マスターからの不正な AXI4-Lite トランザクションを検出します。このとき、接続されているマスターに対してプロトコルに準拠したエラー応答を生成すると共に、この不正なトランザクションが AXI4-Lite スレーブに伝搬されないようにします。
 - AXI4-Lite ソリューションのリソース使用率を最小限に抑えるため、AXI4/AXI3 バーストが AXI4-Lite プロトコル変換によって AXI4-Lite スレーブ向けの一連のシングル ビート トランザクションに変換されることはありません。
 - AXI4-Lite スレーブ デバイスへの書き込みおよび読み出しは単一スレッドのトランザクションです。つまり、一度に 1 つのアドレスのみを伝搬します。これによって通常は、AXI の書き込みと読み出しのアドレス信号を分けることによるリソースのオーバーヘッドを抑止できます。

- ビルトイン AXI3 プロトコル変換
 - AXI3 スレーブ デバイスに接続した場合、AXI インターコネクト コアは、接続されている AXI4 マスターからの 16 ビートを越えるバースト トランザクションを、16 ビート以下の複数のトランザクションに分割します。
- レジスタ スライスによるパイプライン処理 (オプション)
 - マスターやスレーブ デバイスに接続する、各 AXI チャンネルに適用できます。
 - 周波数対レイテンシのトレードオフを調整することで、タイミング クロージャを容易にします。
 - すべての AXI ハンドシェイク条件において、データ スループットを劣化させることなく、1 つのレジスタ スライスあたり 1 レイテンシ サイクルの調整が可能です。
- データパス FIFO によるバッファ処理 (オプション)
 - マスターやスレーブ に接続する、書き込みおよび読み出しのデータパスに適用できます。
 - 深さ 32 の LUT-RAM ベースのバッファを提供します。
 - 深さ 512 のブロック RAM ベースのバッファを提供します。
- 選択可能なインターコネクト アーキテクチャ
 - クロスバー モード (パフォーマンス最適化)
 - 共有アドレス複数データ (SAMD : Shared-Address, Multiple-Data) クロスバー アーキテクチャ
 - 書き込みおよび読み出しデータ チャンネル用の並列のクロスバー パスです。複数の書き込みまたは読み出しデータ ソースが、異なる宛先に対してデータを送信しようとしている場合、AXI の順序の規定に従っていれば、これらのデータ転送を独立して同時に実行できます。
 - コネクティビティ マップのコンフィギュレーションに従った、スパース クロスバー データパスはリソース使用量を低減します。
 - 書き込みアドレスの共有アービタと読み出しアドレスの共有アービタを 1 つずつ含みます。通常、トランザクションが平均 3 データ ビート以上の場合、アービトレーションに要するレイテンシはデータ スループットに影響を与えません。
 - 共有アクセス モード (エリア最適化)
 - 書き込みデータ、読み出しデータおよび単一のアドレス パスを共有します。
 - 一度に 1 つの Outstanding トランザクションを発行します。
 - リソース使用率を最小化します。
- Multiple Outstanding (複数の未処理) トランザクションのサポート (クロスバー モード)
 - 接続されたマスターによる複数の深さレベルの順序変更 (ID スレッド) に対応します。
 - 最大 16 ビット幅の ID 信号に対応します (システム共通)。
 - 書き込み応答の順序変更および読み出しデータの順序変更とインターリーブに対応します。
 - 接続された各マスターに対して、書き込みおよび読み出しトランザクションの受け入れ限度を設定できます。
 - 接続された各スレーブ に対して、書き込みおよび読み出しトランザクションの発行限度を設定できます。
 - オプションのシングル スレッド モード (接続されたマスターごと) は、1 つ以上の Outstanding トランザクションを一度に 1 つのスレッド ID に対してのみ許可することで、スレッド制御のロジックを削減します。
- サイクル依存性 (デッドロック) を回避する「ID あたり 1 スレーブ」の手法
 - 接続されているマスター デバイスが発行した各 ID スレッドについて、インターコネクトは、書き込みおよび読み出しそれぞれに対して一度に各 1 個のスレーブ デバイスのみ、1 つ以上の Outstanding トランザクションを許可します。
- 固定優先順位とラウンドロビン アービトレーション
 - 16 段階の静的優先順位を設定できます。

- より優先順位が高いほかのマスターからの要求がない場合、最低の優先順位に設定された (優先順位 0 の) すべてのマスター間で、ラウンドロビン方式のアービトレーションが実行されます。
- SI スロットのうち受け付け限度に達したもの、発行限度に達した MI スロットを宛先としているもの、またはデッドロックのおそれがある方法で MI スロットにアクセスしようとしているものは、一時的にアービトレーションの対象から外されます。これによって、ほかの SI スロットがアービトレーション可能になります。
- 接続された各スレーブに対する TrustZone セキュリティの一括サポート
 - セキュアなスレーブ デバイスとして設定すると、セキュアな AXI アクセスしか受け入れなくなります。
 - セキュアでないアクセスはブロックされ、AXI インターコネクト コアがマスター デバイスに対して DECERR 応答を返します。
- リソース使用量を低減する読み出し専用および書き込み専用マスター/スレーブのサポート

CORE Generator ツールでサポートされる機能

CORE Generator ツール フローでサポートされる AXI インターコネクト コアの機能は、次のとおりです。

- AXI プロトコル (AXI4 のみ) に準拠した次の機能
 - 最大バースト長 256 のインクリメンタル (INCR) バーストが可能です。
 - サービス品質 (QoS) 信号が存在する場合、それを伝搬します。この信号は AXI インターコネクト コアでは使用しません (オプション)。
- インターフェイス データ幅
 - 32、64、128、256、512、1024 ビット
- アドレス幅: 12 ~ 64 ビット
- 最大 16 個のスレーブ インターフェイス (接続された最大 16 個のマスター デバイスからのトランザクションを受け入れ) と 1 個のマスター インターフェイス (接続された 1 個のスレーブ デバイスにトランザクションを発行)
 - マスターとスレーブを 1 対 1 で接続する場合、AXI インターコネクト コアは、データ幅変換、クロック レート変換、レジスタのパイプライン処理、データパスのバッファ処理などのオプション機能も実行可能です。
- ビルトイン データ幅変換
 - マスターおよびスレーブの各接続では、それぞれ独立したデータ幅 (32、64、128、256、512、1024 ビットのいずれか) を使用できます。
 - 内部クロスバーは、ネイティブ データ幅が 32、64、128、256、512、1024 ビットになるようなコンフィギュレーションが可能です。
 - クロスバーのネイティブ データ幅と異なる幅のマスターおよびスレーブ接続ではデータ幅を変換します。
 - よりデータ幅の広いインターフェイスに変換する場合 (アップサイジング)、データをパック (結合) します。それには、アドレス チャネル制御信号によって許可する (CACHE 変更許可ビットをアサートする) 必要があります。
 - よりデータ幅の狭いインターフェイスに変換 (ダウンサイジング) する際に、そのままでは最大バースト長を超えてしまう場合は、バースト トランザクションを複数のトランザクションに分割できます。
- ビルトイン クロック レート変換
 - マスターおよびスレーブの接続は、それぞれ独立したクロック レートを使用できます。
 - 内部クロスバーのネイティブ クロック レートへ整数比 (N:1 および 1:N) の同期変換が可能です。
 - 非同期クロック変換にも対応しています (ただし、同期変換よりも多くのストレージを使用し、より長いレイテンシが発生)。
 - AXI インターコネクト コアは、接続された各マスター/スレーブ ペアのクロック レートに再同期されたリセット信号をエクスポートします。

- レジスタ スライスによるパイプライン処理 (オプション)
 - マスターやスレーブ デバイスに接続する AXI チャネルすべてで使用できます。
 - 周波数対レイテンシのトレードオフを調整することで、タイミング クロージャを容易にします。
 - すべての AXI ハンドシェイク条件において、データ スループットを劣化させることなく、1 つのレジスタ スライスあたり 1 レイテンシ サイクルの調整が可能です。
- データパス FIFO によるバッファ処理 (オプション)
 - マスターやスレーブに接続する、書き込みおよび読み出しのデータパスに適用できます。
 - 深さ 32 の LUT-RAM ベースのバッファを提供します。
 - 深さ 512 のブロック RAM ベースのバッファを提供します。
 - バースト中のフル/エンプティによるストールを回避するために、パケット FIFO 動作オプションを提供します。
- Multiple Outstanding (複数の未処理) トランザクションのサポート
 - 接続されたマスターによる複数の深さレベルの順序変更 (ID スレッド) に対応します。
 - 接続された各マスター デバイスからの最大 8 ビット幅の ID 信号に対応します (最大 12 ビット幅の ID 出力を生成)。
 - 書き込み応答の順序変更および読み出しデータの順序変更とインターリーブに対応します。
 - 接続された各マスターに対して、書き込みおよび読み出しトランザクションの受け入れ限度を設定できます。
 - 接続されたスレーブに対して、書き込みおよび読み出しトランザクションの発行限度を設定できます。
- 固定優先順位とラウンドロビン アービトレーション
 - 16 段階の静的優先順位を設定できます。
 - より優先順位が高いほかのマスターからの要求がない場合、最低の優先順位に設定された (優先順位 0 の) すべてのマスター間で、ラウンドロビン方式のアービトレーションが実行されます。
 - 受け入れ限度に達したマスター デバイスは、一時的にアービトレーションの対象から外されます。これによって、接続されているほかのマスターがアービトレーション可能になります。
- リソース使用率を低減する読み出し専用および書き込み専用マスター デバイスのサポート

CORE Generator ツール フローの制約のまとめ

ここで解説している機能について、CORE Generator ツール フローでは次の制約があります。

- アドレス マップおよびアドレス デコード関連の全機能を定義するプロセス、複数の宛先スレーブ デバイスを選択するプロセスは、現時点では XPS フロー用に予約されています。CORE Generator ツール フローで使用了場合、インターコネクトは 1 個のスレーブ デバイスのみへの接続をサポートし、すべてのトランザクション アドレスは単純に伝搬されます。また、インターコネクトにデコード エラー (DECERR) 応答を生成させる条件はありません。
- AXI4 (メモリマップ方式) プロトコルのみがサポートされています。CORE Generator ツール フローで使用する場合、インターコネクトは、外部メモリ コントローラーなどのメモリ タイプのスレーブ デバイスのみへの接続用です。制御レジスタ (AXI4-Lite) スレーブへは接続できません。
- USER 信号はサポートされていません。
- レジスタ スライスは、インターフェイスごとに選択でき、有効にした場合、すべての AXI チャネルに適用されます。チャネルごとに使用されるレジスタ スライスのタイプは、W および R チャネルでは完全レジスタ化、AW、AR、および B チャネルでは軽量化に固定されています。
- 完全なクロスバー モードのみがサポートされており、アドレス転送とデータ転送の両方について、独立した書き込みおよび読み出し動作が可能です。「共有アクセス」モードはサポートされていません。接続されるスレーブ デバイスが 1 個のみのため、スパス クロスバー接続は適用されません。

- ID 信号の幅はグローバルに選択され、すべてのスレーブ インターフェイス (SI) ポートに適用されます。インターコネクトは、マスター インターフェイス ID 信号を発行する際、常に、生成元の SI インデックス番号を示す 4 つの上位ビットを付加します。
- TrustZone セキュリティは、インターコネクトのサービスとして提供されません (AW/ARPROT 信号は、スレーブ デバイスに伝搬される)。

AXI インターコネクト コアの制約

次の制約は、AXI インターコネクト コア全般に適用されます (XPS と CORE Generator ツール フローの両方)。

- AXI インターコネクト コアは、次の AXI3 機能には対応していません。
 - アトミック ロック トランザクション。この機能は AXI4 プロトコルでは廃止されました。ロックされたトランザクションは、ロックを解除されたトランザクションに変換されたうえで MI によって伝搬されます。
 - 書き込みインターリーブ。この機能は AXI4 プロトコルでは廃止されました。したがって、AXI3 マスター デバイスは、書き込みインターリーブの深さが 1 のスレーブに接続するように設定する必要があります。
- AXI4 の QoS 信号は、アービトレーションの優先順位に影響を与えません。QoS 信号は、SI から MI に伝搬されます。
- 複数ビートのバーストを複数の単一ビート トランザクションに変換する AXI インターコネクト コアの機能は、AXI4-Lite スレーブ デバイスを宛先とした場合は実行されません。
- AXI インターコネクト コアは低消費電力モードまたは AXI C チャンネル信号の伝搬には対応していません。
- AXI インターコネクト コアは、AXI チャンネル伝送のいずれかの宛先が永久にストールしても、タイムアウトしません。したがって、接続されたすべての AXI スレーブは受信したトランザクションのすべてに対して AXI プロトコルの要件に従った応答を返す必要があります。
- AXI インターコネクト コアはアドレス マップの再割り当てを行いません。
- AXI インターコネクト コアは、APB などの非 AXI プロトコルへのビルトイン変換機能を備えていません。
- AXI インターコネクト コアにはクロック イネーブル (ACLKEN) 入力がありません。このため、ザイリンクス システムではメモリマップ方式の AXI インターフェイス間で ACLKEN を使用できません。

注記：ザイリンクス AXI4-Stream インターフェイスは ACLKEN 信号をサポートしています。

定義、頭字語、略語

表 1 に、この文書で使用されている頭字語、略語、定義の一覧を示します。

表 1：定義、頭字語、略語

項目	説明
AXI	インプリメントされる AXI プロトコル インターフェイス全般を指す一般的な用語。
マスター デバイス または接続された マスター	AXI トランザクションを生成してスレーブ IP へと接続されたワイヤに発信する、IP またはデバイス (または IP 上の複数のインターフェイスのうちの 1 つ)。
スレーブ デバイス または接続された スレーブ	マスター IP へと接続されたワイヤから AXI トランザクションを受信して応答する、IP またはデバイス (または IP 上の複数のインターフェイスのうちの 1 つ)。
マスター インター フェイス (汎用)	アウトバウンド AXI トランザクションを生成して AXI 転送の開始側 (送信元) となる IP またはモジュールのインターフェイス。AXI のマスター インターフェイスでは、AWVALID、ARVALID、WVALID が出力、RVALID、BVALID が入力。
スレーブ インター フェイス (汎用)	インバウンド AXI トランザクションを受信して AXI 転送の受信側 (宛先) となる IP またはモジュールのインターフェイス。AXI のスレーブ インターフェイスでは、AWVALID、ARVALID、WVALID が入力、RVALID、BVALID が出力。
SI	AXI インターコネクトのスレーブ インターフェイス：XPS フローでは、接続されたすべてのマスター デバイスからのインバウンド AXI トランザクションを受信する、ベクター化された AXI スレーブ インターフェイス。CORE Generator ツール フローでは、1 つのマスター デバイスに接続する複数のスレーブ インターフェイスの 1 つ。
MI	AXI インターコネクトのマスター インターフェイス：XPS フローでは、接続されたすべてのスレーブ デバイスへのアウトバウンド AXI トランザクションを生成する、ベクター化された AXI マスター インターフェイス。CORE Generator ツール フローでは、1 つのスレーブ デバイスに接続する 1 つのマスター インターフェイス。
SI スロット	スレーブ インターフェイス スロット：1 つのマスター デバイスに接続する AXI インターコネクト コアのスレーブ インターフェイス ベクター信号の 1 スライス。
MI スロット	マスター インターフェイス スロット：1 つのスレーブ デバイスに接続する AXI インターコネクト コアのマスター インターフェイス ベクター信号の 1 スライス。
SI 側	AXI インターコネクト コアの SI 側に近いモジュール インターフェイス。
MI 側	AXI インターコネクト コアの MI 側に近いモジュール インターフェイス。
クロスバー	AXI インターコネクト コアの中心に位置し、さまざまな SI スロットおよび MI スロット間のアドレス、データ、応答の各チャネルの転送をルーティングするモジュール。
SI 域	SI とクロスバーの間に配置された、AXI インターコネクト コアの変換およびストレージ モジュール群。
MI 域	クロスバーと MI の間に配置された、AXI インターコネクト コアの変換およびストレージ モジュール群。
アップサイザー	SI 側から MI 側へとデータを移動する際に (書き込み/読み出しの方向に関係なく)、データパス幅を広くする変換機能。
ダウンサイザー	SI 側から MI 側へとデータを移動する際に (書き込み/読み出しの方向に関係なく)、データパス幅を狭くする変換機能。

ファンクションの説明

図 1 に、AXI インターコネクト コアの最上位のブロック図を示します。

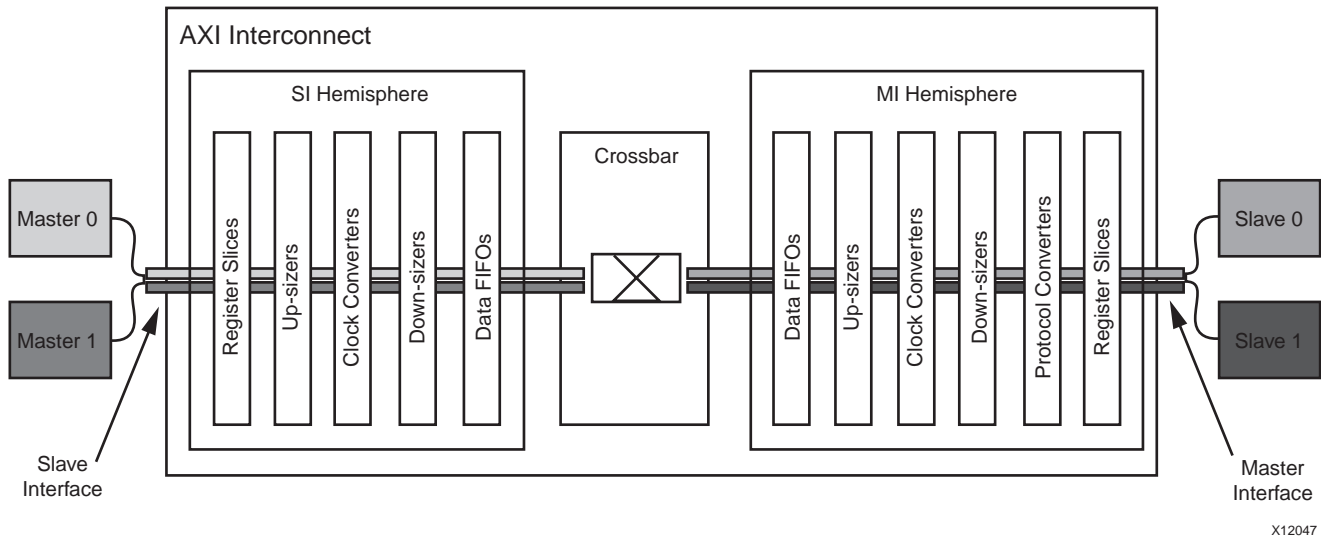


図 1：AXI インターコネクト コア接続図

AXI インターコネクト コアは、SI、MI と、その間を結ぶ AXI チャネルパスを構成するファンクション ユニットから成ります。SI は、接続されたマスター デバイスからの書き込みおよび読み出しトランザクション要求を受け入れます。MI はスレーブ デバイスに対してトランザクションを発行します。中央に位置するクロスバーは、SI と MI に接続されたさまざまなデバイス間のすべての AXI チャネルにトラフィックを振り分けます。AXI インターコネクト コアには、クロスバーと各インターフェイス間に配置され、さまざまな変換およびストレージ機能を果たすファンクション ユニットも含まれます。クロスバーは、SI 関連のファンクション ユニット (SI 域) と MI 関連のファンクション ユニット (MI 域) の中央で、AXI インターコネクト コアを 2 つに分割しています。

次に、AXI インターコネクト コアの使用モデルについて説明します。

使用モデル

AXI インターコネクト コアは、1 つ以上のメモリマップ方式 AXI マスター デバイスと、1 つ以上のメモリマップ方式のスレーブ デバイスを接続します。その使用例について説明します。

- パススルー
- 変換のみ
- N 対 1 のインターコネクト
- 1 対 N のインターコネクト
- N 対 M のインターコネクト (クロスバー モード)
- N 対 M のインターコネクト (共有アクセス モード)

パススルー

AXI インターコネクト コアに接続されるデバイスがマスターとスレーブのそれぞれ 1 つだけで、オプションの変換機能やパイプライン処理を実行しない場合、スレーブ インターフェイスとマスター インターフェイス間のすべてのパスはワイヤによる直接接続になり、レイテンシを発生せず、ロジック リソースも消費しません。図 2 はパススルーの接続図です。ただし、ARESET_OUT_N 出力に接続しているマスターまたはスレーブ デバイスでは、AXI インターコネクト コアは、INTERCONNECT_ARESETN 入力をスレーブおよびマスター インターフェイスそれぞれのクロック ドメインに再同期し続けます。これにより、少数のフリップフロップが消費されます。

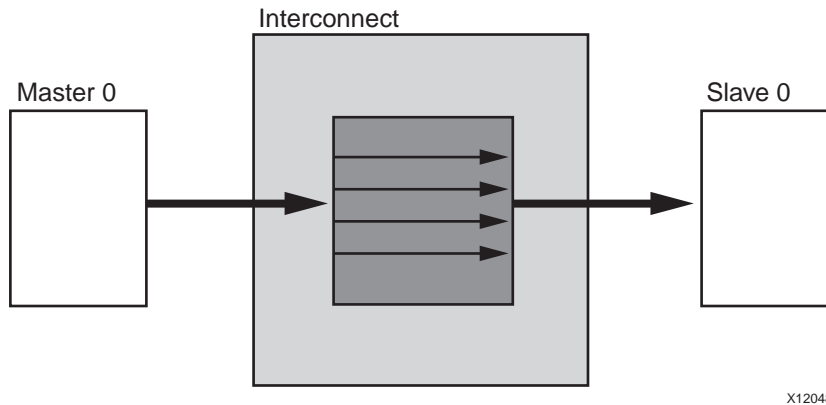


図 2 : AXI インターコネクトの使用例 – パススルー

変換のみ

AXI インターコネクト コアは、1 つのマスターを 1 つのスレーブに接続した場合、次のようなさまざまな変換およびパイプライン機能を実行できます。

- データ幅変換
- クロック レート変換
- AXI4-Lite スレーブ適合
- AXI-3 スレーブ適合
- レジスタ スライスまたはデータ チャネル FIFO によるパイプライン処理

これらの使用例では、AXI インターコネクト コアにアービトレーション、デコード、ルーティングのロジックは含まれません (オプションのアドレス範囲チェックが無効の場合に限る)。また、実行する変換によってはレイテンシが発生する場合があります。図 3 は、1 対 1 変換の使用例を示したものです。

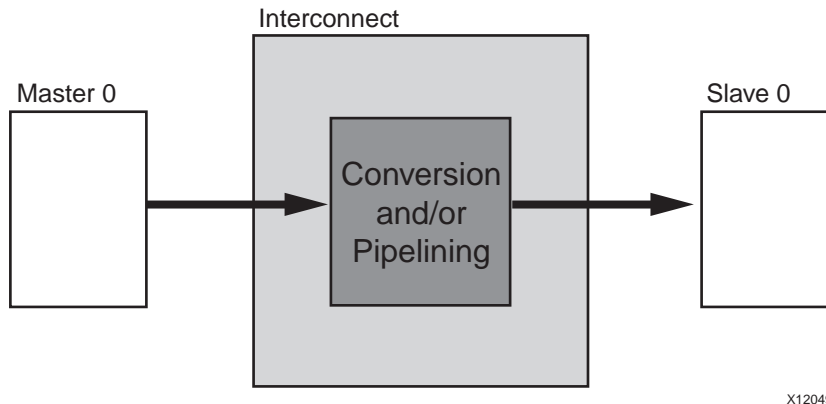


図 3 : AXI インターコネクトの使用例 – 1 対 1 変換

N 対 1 のインターコネクト

一般に、このような AXI インターコネクト コアのコンフィギュレーションを行うのは、複数のマスター デバイスが、1 つのスレーブ デバイス、通常はメモリ コントローラーへのアクセスについてアービトレーションを実行する場合です。

このようなケースでは、アドレス デコード ロジックが不要になることがあり、その場合 AXI インターコネクト コアから除外されます (オプションのアドレス範囲検証が無効の場合に限る)。図 4 に示すように、このコンフィギュレーションでもデータ幅やクロック レートなどの変換機能はすべて動作します。

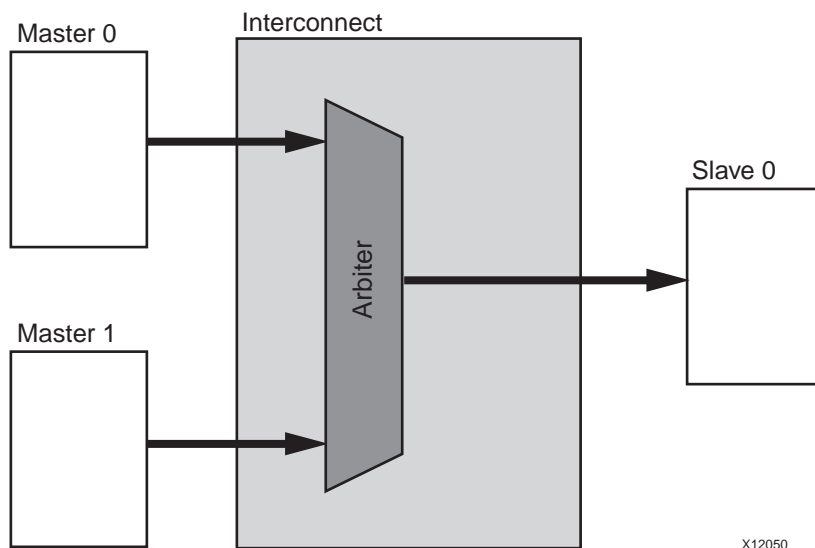


図 4 : AXI インターコネクトの使用例 – N 対 1

1 対 N のインターコネクト

もう 1 つの AXI インターコネクト コアのコンフィギュレーション ケースが、1 つのマスター デバイス、通常はプロセッサが、複数のメモリマップ方式のスレーブ ペリフェラルにアクセスする場合です。このような場合、図 5 に示すようにアドレスおよび書き込みデータパスのアービトレーションは実行されません。

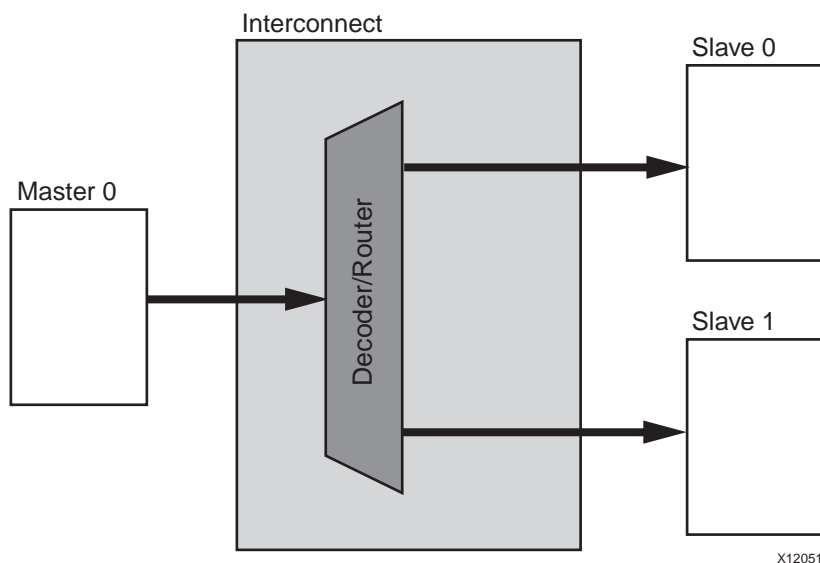


図 5 : AXI インターコネクトの使用例 – 1 対 N

N 対 M のインターコネクト (クロスバー モード)

クロスバー モードにおける N 対 M の AXI インターコネクト コアでは、SAMD (共有アドレス複数データ) トポロジを取ることができます。これは、図 6 および 図 7 に示すようにスパス データ クロスバー接続から構成され、単スレッドの共有書き込みおよび読み出しアドレス アービトレーションが実行されます。

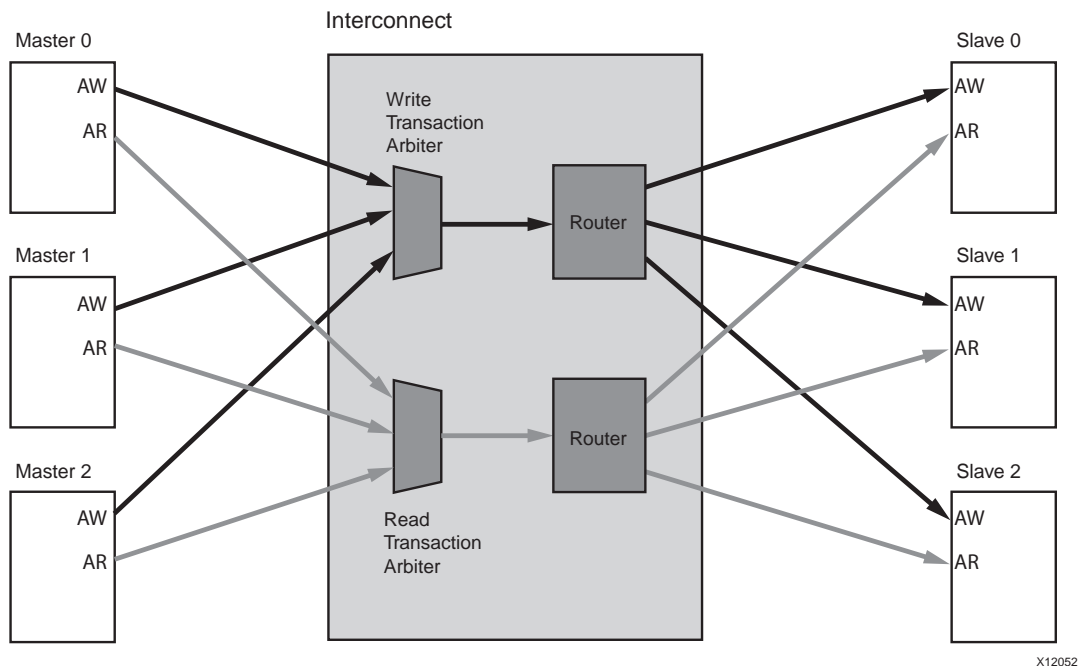


図 6 : 共有書き込みおよび読み出しアドレス アービトレーション

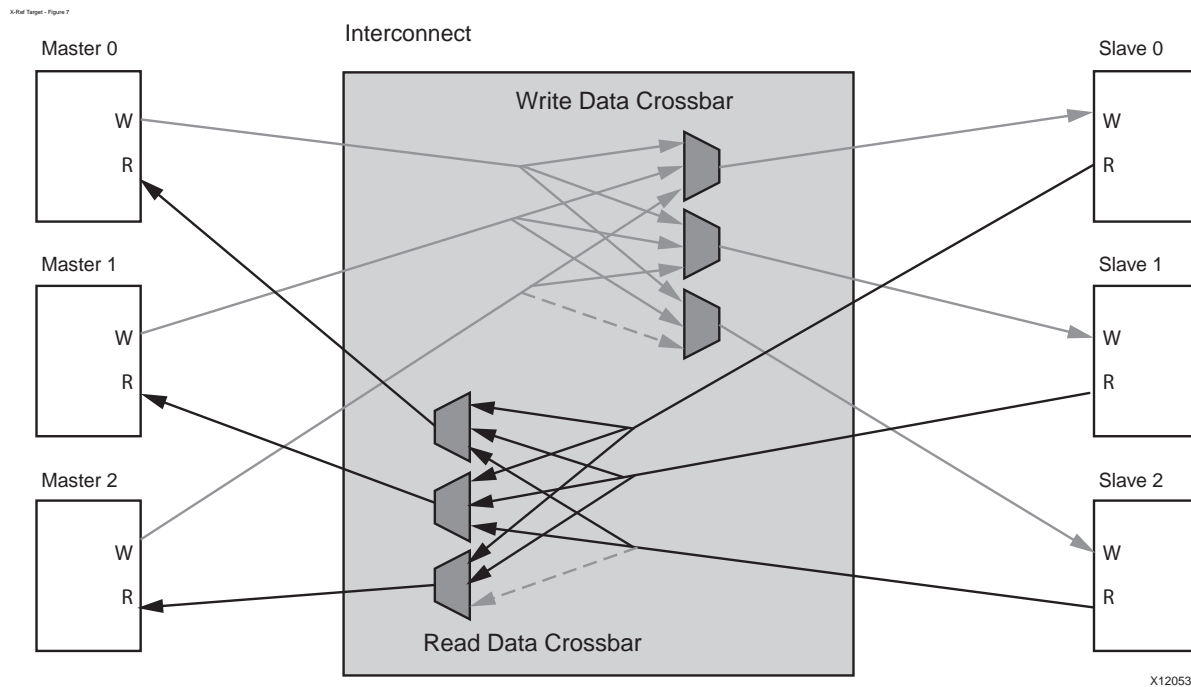


図 7 : スパス クロスバーによる書き込みおよび読み出しデータパス

並列書き込みおよび読み出しデータパスは、スパース コネクティビティ マップのコンフィギュレーションに従い、各 SI スロットを、アクセス可能なすべての MI スロットに接続します。複数のソースが異なる宛先に対してデータを送信しようとしている場合、AXI の順序の規定に従っていれば、データ転送を独立して同時に実行できます。

すべての SI スロット間の書き込みアドレス チャンネルはセントラル アドレス アービタに入力され、アービタは一度に 1 つの SI スロットへのアクセスを許可します。読み出しアドレス チャンネルの場合も同様です。

各アービトレーション サイクルでアクセス権を得たマスターは、宛先の MI スロットにアドレス情報を転送し、該当するコマンド キューにエントリをプッシュします。このようにして AXI の順序の規定に従いつつ、さまざまなデータパスを経由して適切な宛先にデータが送られます。

N 対 M のインターコネクト (共有アクセス モード)

図 8 に示すように、共有アクセス モードにおける N 対 M 接続の場合、AXI インターコネクト コアは一度に 1 つだけ Outstanding トランザクションを提供します。接続された各マスターでは、読み出しトランザクション要求が書き込み要求よりも常に優先されます。次に要求を送信したマスターのいずれかをアービタが選択します。宛先のスレーブ デバイスに対する書き込みまたは読み出しデータ転送が有効になります。データ転送が完了 (書き込み応答を含む) すると、次の要求が選択され処理されます。共有アクセス モードはインターコネクトのクロスバー モジュールをインプリメントするために必要なリソースを最小限に抑えます。

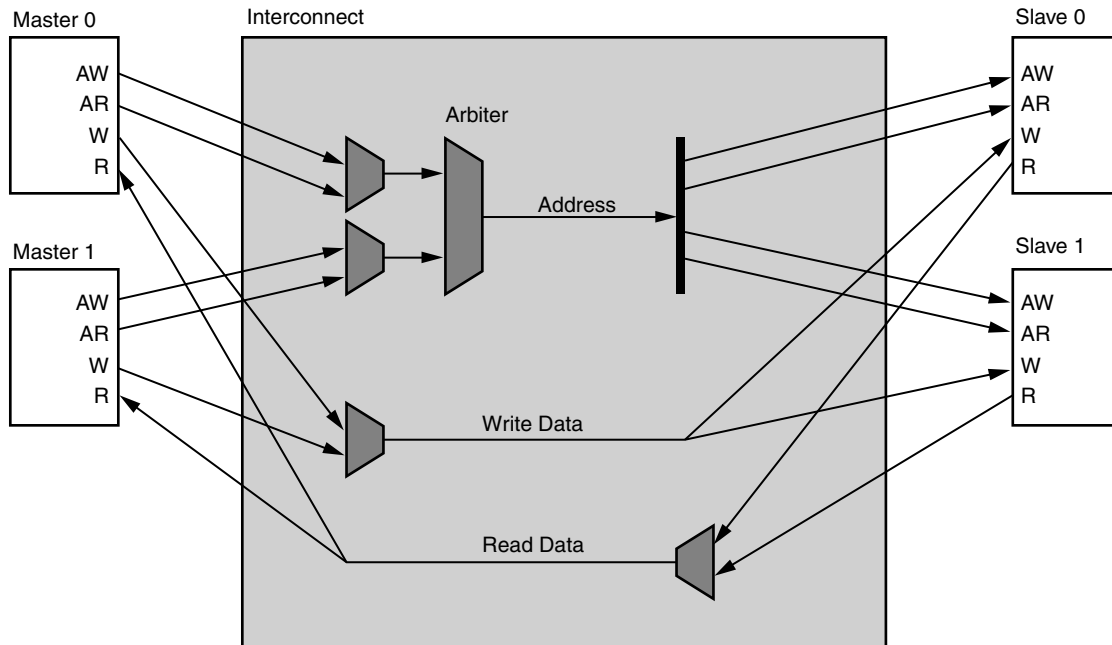


図 8 : 共有アクセス モード

AXI インターコネクト コアの機能

以降のサブセクションでは、AXI インターコネクト コアの機能を説明します。

- 最上位のスレーブ/マスター インターフェイス
- 幅変換
- 幅変換トランザクション
- クロック変換
- ペリフェラル レジスタ スライス
- データパス FIFO

- [ID 信号の使用方法](#)
- [複数アドレス範囲のサポート](#)
- [サイクル依存性の回避](#)
- [エラー信号](#)

最上位のスレーブ/マスター インターフェイス

XPS フローを使用した場合、最上位のインターフェイスは、それぞれベクター化された単一の AXI SI と AXI MI で構成されます。

ベクター化された各インターフェイスはコンフィギュレーション可能であり、1 から 16 までのマスター/スレーブ デバイスに接続します。

コア上のベクター化された AXI インターフェイスの各信号幅は、元の信号幅に、接続されたデバイス数を乗算した値になります。1 つのデバイスに接続されたビット スライスのすべてを、インターフェイスの 1 つのスロットといいます。たとえば、AWLEN 信号が書き込みトランザクションのデータ ビート数を示す 8 ビットの値を保持しています。AXI インターコネクト コアが 2 つの SI スロットで構成される場合、S_AXI_AWLEN 信号の幅は合計 16 ビットになります。

WDATA、WSTRB、RDATA 信号の有効幅も、MI または SI 単位に設定できます。ベクター化された SI または MI 上のこれら各信号の幅は、すべての SI および MI スロットに設定された信号幅、インターコネクトのネイティブ データ幅のうちの最大値にスロット数を乗算したものになります。それよりも幅の狭いスロットの未使用の上位ビットは、AXI インターコネクト コア内で固定されるか (入力の場合)、または未接続のまま (出力の場合) となり、インプリメンテーション ツールで切り捨てられます。このように、AXI インターフェイスの各信号は全スロット同じ物理幅で割り付けられます。

たとえば、[図 9](#) に示すように、AXI インターコネクト コアが 2 つの SI スロットで構成されていて、1 つのデータ幅が 32 ビット、もう 1 つのデータ幅が 128 ビットで、MI スロットまたはインターコネクトのいずれにもこれより大きなデータ幅が設定されていない場合、SI の WDATA 信号および RDATA 信号の幅は合計 256 ビットになります。

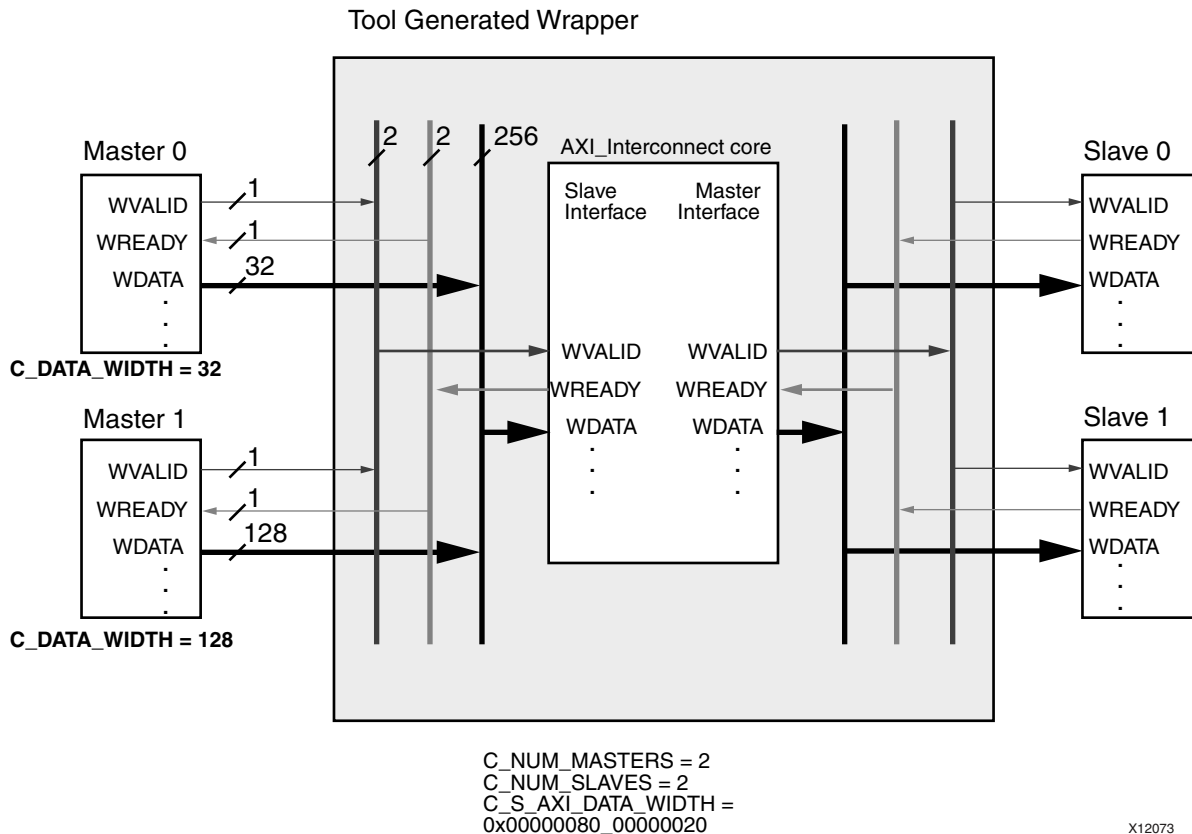


図 9 : ベクター化されたスレーブ/マスター インターフェイス

特に次の点に注意してください。

- 「スロット 0」は WDATA[31:0] を使用します。
- 「スロット 1」は WDATA[255:128] を使用し、「マスター 1」デバイスの WDATA[127:0] に接続します。
- WDATA[127:32] は AXI インターコネクト コア内で固定されるか、または未接続のままとなります。

I/O 信号と同様に、AXI インターコネクト コアの多くのコンフィギュレーション パラメーターもすべての SI スロット、MI スロットでベクターとしてフォーマットされます。ベクター化されたパラメーターは、次のようにフォーマットされます。

- TrustZone セキュリティ インジケーター (C_M_AXI_SECURE) など、ブール条件を定義するパラメーターは、1 スロットあたり 1 ビットのビット ベクターとしてフォーマットされます。
- 数値を定義するパラメーターは、値の範囲にかかわらず、1 スロットあたり 32 ビットのビット ベクターとしてフォーマットされます。
- ベース アドレスと高次アドレスは例外であり、1 スロットあたり 64 ビットとしてフォーマットされます。

図 9 の例で、SI (C_S_AXI_DATA_WIDTH) 上のスロットの有効データ幅を定義するベクター化されたパラメーターの値は 0x0000008000000020 となります。ここで、0x20 はスロット 0 が 32 ビットであることを、0x80 はスロット 1 が 128 ビットであることを表しています。パラメーター値は I/O 信号と同様にリトル エンディアンであり、したがってスロット 0 に対応する値はパラメーター ベクターの右端の最下位ビット (LSB) に表示されます。

CORE Generator ツール フローを使用した場合、ベクター化されたインターフェイス上に、1 つのモジュール レイヤーが挿入されます。このモジュール レイヤーは、ベクター化されたスレーブ インターフェイスを、HDL デザインで AXI マスター デバイスに直接接続するのに適した個々の列挙型のインターフェイスに分割します。この最上位モジュールは、SI 関連の各パラメーターも同様に分割します。

幅変換

AXI インターコネクト コアには、パラメーターによって内部ネイティブ データ幅が定義されています。サポートされるデータ幅は 32、64、128、256、512、1024 ビットです。クロスバーに接続された AXI データ チャンネルは、C_INTERCONNECT_DATA_WIDTH パラメーターで指定される AXI インターコネクト コアのネイティブ幅にサイズ調整されます。

これとは異なる幅の SI スロットまたは MI スロットがあると、AXI インターコネクト コアは幅変換ユニットを挿入して、スロット幅を AXI インターコネクト ネイティブ幅に適合させてから、クロスバーをもう一方の側へと横断させます。

データが SI から MI に向かう方向で見てデータパス幅が広がる (アップサイジング) 場合と、狭くなる (ダウンサイジング) 場合で、幅変換の機能は異なりますが、SI 域 (SI から AXI インターコネクト ネイティブ幅への変換) と MI 域 (AXI インターコネクト ネイティブ幅から MI への変換) では同じになります。

MI と SI の各スロットには、パラメーターによって個別にデータ幅の値が関連付けられています。AXI インターコネクト コアは、MI および SI の各スロットを次の手順で自動的に内部ネイティブ データ幅に適合させます。

- SI スロットのデータ幅が AXI インターコネクト コアの内部ネイティブ データ幅よりも広い場合、SI スロットのパスに沿ってダウンサイジングの変換を実行します。
- AXI インターコネクト コアの内部ネイティブ データ幅が MI スロットのデータ幅よりも広い場合、MI スロットのパスに沿ってダウンサイジングの変換を実行します。
- SI スロットのデータ幅が AXI インターコネクト コアの内部ネイティブ データ幅よりも狭い場合、SI スロットのパスに沿ってアップサイジングの変換を実行します。
- AXI インターコネクト コアの内部ネイティブ データ幅が MI スロットのデータ幅よりも狭い場合、MI スロットのパスに沿ってアップサイジングの変換を実行します。

以降のサブセクションで、ダウンサイジングとアップサイジングについて説明します。

ダウンサイジング

SI 側のデータ幅が MI 側よりも広く、かつトランザクションの転送サイズも MI 側のデータ幅より広い場合、ダウンサイジングが実行され、MI 側に発行されるトランザクションのデータ ビート数は分割数に応じて倍増します。

- 書き込みの場合、データがシリアル化されます。
- 読み出しの場合、データが結合されます。

AXI インターコネクト コアは (SI 上の) 各出力データ ビートの RRESP を、結合した入力データ ビート内で発生したワースト エラー条件に設定します。その場合の優先順位は高い方から DECERR、SLVERR、OKAY、EXOKAY です。

トランザクションの転送サイズが MI 側のデータ幅以下であれば、そのトランザクション (アドレス チャンネルの値) は変更されません。転送データはバイト レーン ステアリングの場合を除き、そのまま通過します。これは、書き込みおよび読み出しの両方に当てはまります。

ダウンサイジングの際には AXI インターコネクト コアは各バーストの長さを計算し、累計バースト長がバースト上限 (AXI4 で 256 データ ビート) を超えそうになると検知します。その場合、AXI インターコネクト コアは、適合するサイズの複数のバースト トランザクションに自動的にトランザクションを分割します。

- AWLOCK または ARLOCK 信号が、排他アクセスの書き込みまたは読み出しトランザクションを示している場合に、ダウンサイジングによって分割が実行されると、AXI インターコネクト コアはすべての出力トランザクションの LOCK 信号を通常アクセス (0) を示すように変更します。
- ダウンサイジングされた書き込みトランザクションが分割された場合、AXI インターコネクト コアは MI における複数の書き込み応答をまとめ、1 つの書き込み応答を SI で発行します。コアは、エラー応答コード (BRESP) を複数の入力応答内で発生したワースト ケースのエラー条件に設定します。その場合の優先順位は高い方から DECERR、SLVERR、OKAY です (分割されたトランザクションでは EXOKAY は発生しない)。

トランザクションの分割を含むダウンサイジングは、AW/ARCACHE 信号の値 (具体的には「変更許可」ビット) によって制限されません。ダウンサイジングによるトランザクションの分割は、トランザクションを完了するほかの方法がないため、CACHE 信号によって制限できません。さまざまなサイズ変換については、[18 ページの表 2](#) を参照してください。

ダウンサイザーのモジュールによって複数の **Outstanding** トランザクションを伝搬できます。AW/AR チャネル転送からのトランザクション特性として、対応する応答転送を待機する間、キューに入ります。ただし、書き込み応答または読み出しデータの順序が変更する可能性があるため、AW および AR チャネル ダウンサイザーによるトランザクションの受け入れは一度に 1 つの ID スレッドに限られます。

インターコネクトは、SI とクロスバー間、クロスバーと MI 間のいずれにおいても、1024 ビットから 32 ビットへの直接のダウンサイジングはサポートしていません。いずれかの SI が 1024 ビットの場合、C_INTERCONNECT_DATA_WIDTH は 32 より大きい値に設定します。MI が 32 ビットの場合の C_INTERCONNECT_DATA_WIDTH には、1024 より小さい値を設定します。

アップサイジング

MI 側のデータ幅が SI 側よりも広い場合はアップサイジングが実行されます。AW/ARCACHE[1] ビット (変更許可) がアサートされていると、データの圧縮が実行されます (INCR バーストおよび WRAP バーストの場合)。

その結果、MI 側に発行されるトランザクションのデータ ビート数は圧縮に応じて減少します。

- 書き込みの場合、データが結合されます。
- 読み出しの場合、データがシリアル化されます。
 - AXI インターコネクト コアは各入力データ ビートの RRESP を、各出力データ ビート (SI 上) の RRESP として複製します。

AW/ARCACHE[1] ビットがディアサートされると、トランザクション (アドレス チャネルの値) は変化せず、転送データはバイト レーン ステアリングの場合を除き、そのまま通過します。

この後者の機能は、エクспанダーと呼ばれることがあります。

アップサイジングでトランザクションが分割されることはありません。さまざまなサイズ変換については、[表 2](#) を参照してください。

アップサイザーのモジュールによって複数の **Outstanding** トランザクションを伝搬できます。AW/AR チャネル転送からのトランザクション特性として、対応する応答転送を待機する間、キューに入ります。ただし、読み出しデータの順序が変更する可能性があるため、AR チャネル アップサイザーによるトランザクションの受け入れは一度に 1 つの ID スレッドに限られます。B チャネル応答はアップサイザーによる変換が不要であり、受信したままの任意の順序で伝搬できるため、書き込みトランザクションは ID スレッドによる制約を受けません。

幅変換トランザクションの変化

[表 2](#) では、特性、信号、および導出される式に次の記号を使用しています。

- si = スレーブ インターフェイス (SI)
- cb = インターコネクト (クロスバー) コア
- mi = マスター インターフェイス (MI)

[表 2](#) には次の内容を一覧表示しています。

- 相対的な DWidth を si.DW と cb.DW で比較した場合の SI 域の変化
- 相対的な DWidth を cb.DW と mi.DW で比較した場合の MI 域の変化

根拠となる式

[表 2](#) は、幅変換の式を列挙したものです。

1. 幅変換によってトランザクションの長さが変更される場合、出力 SIZE は常に出力 DATA_WIDTH と同じです。

2. $si.DW = C_S_AXI_DATA_WIDTH$
3. $cb.DW = C_INTERCONNECT_DATA_WIDTH$
4. $mi.DW = C_M_AXI_DATA_WIDTH$
5. $si.Bytes = si.DW^{[2]} / 8$
6. $cb.Bytes = cb.DW^{[3]} / 8$
7. $mi.Bytes = mi.DW^{[4]} / 8$
8. $cb.ByteMask = cb.Bytes^{[5]} - 1$
9. $mi.ByteMask = mi.Bytes^{[6]} - 1$
10. $si.SIZE = S_AXI_AWSIZE$ または S_AXI_ARSIZE の当てはまる方
11. $cb.SIZE = si.SIZE$ if $(cb.LEN=si.LEN)$, else $\log_2(cb.Bytes)^{[6]}$
12. $mi.SIZE = cb.SIZE$ if $(mi.LEN=cb.LEN)$, else $\log_2(mi.Bytes)^{[7]}$
13. $si.SizeMask = (2^{**}si.SIZE^{[10]}) - 1$
14. $cb.SizeMask = (2^{**}cb.SIZE^{[11]}) - 1$
15. $mi.SizeMask = (2^{**}mi.SIZE^{[12]}) - 1$
16. $cb.AlignedStart = si.ADDR \& \sim cb.ByteMask^{[8]}$
17. $cb.AlignedEnd = ((si.ADDR \& \sim si.SizeMask^{[13]}) + (si.LEN * 2^{**}si.SIZE^{[10]})) \& \sim cb.ByteMask^{[9]}$
18. $cb.upsize_LEN = (cb.AlignedEnd^{[17]} - cb.AlignedStart^{[16]}) / cb.Bytes^{[6]}$
19. $mi.AlignedStart = cb.ADDR \& \sim mi.ByteMask^{[9]}$
20. $mi.AlignedEnd = ((cb.ADDR \& \sim cb.SizeMask^{[13]}) + (cb.LEN * 2^{**}cb.SIZE^{[11]})) \& \sim mi.ByteMask^{[9]}$
21. $mi.upsize_LEN = (mi.AlignedEnd^{[20]} - mi.AlignedStart^{[19]}) / mi.Bytes^{[4]}$
22. $si.conv_ratio = (2^{**}si.SIZE^{[10]}) / cb.Bytes^{[8]}$
23. $cb.conv_ratio = (2^{**}cb.SIZE^{[10]}) / mi.Bytes^{[9]}$
24. $si.downsize_LEN = (si.LEN+1) * si.conv_ratio - 1^{[22]}$
25. $cb.downsize_LEN = (cb.LEN+1) * cb.conv_ratio - 1^{[23]}$
26. $cb.AlignedAdjustment = (si.ADDR \& si.SizeMask^{[13]} \& \sim cb.ByteMask^{[8]}) / cb.Bytes^{[6]}$
27. $mi.AlignedAdjustment = (cb.ADDR \& cb.SizeMask^{[14]} \& \sim mi.ByteMask) / mi.Bytes^{[9]}$
28. $si.burst_bytes = 2^{**}si.SIZE^{[10]} * (si.LEN+1)$
29. $cb.burst_bytes = 2^{**}cb.SIZE^{[11]} * (cb.LEN+1)$
30. $si.burst_mask = si.burst_bytes^{[28]} - 1$
31. $cb.burst_mask = cb.burst_bytes^{[29]} - 1$
32. $si.wrap_address = si.ADDR \& \sim si.burst_mask^{[30]}$
33. $cb.wrap_address = cb.ADDR \& \sim cb.burst_mask^{[31]}$
34. $si.wrap1_LEN = (si.burst_bytes^{[28]} - (si.ADDR \& si.burst_mask^{[30]})) / cb.Bytes - 1^{[8]}$
35. $cb.wrap1_LEN = (cb.burst_bytes^{[29]} - (cb.ADDR \& cb.burst_mask^{[31]})) / mi.Bytes - 1^{[7]}$
36. $si.wrap2_LEN = (si.ADDR \& si.burst_mask^{[30]}) / cb.Bytes - 1^{[6]}$
37. $cb.wrap2_LEN = (cb.ADDR \& cb.burst_mask^{[31]}) / mi.Bytes - 1^{[7]}$

注記: 「x%y」は x を y で割った余りを表します。

幅変換トランザクション

表 2：幅変換トランザクション

相対 DWidth	条件	出カトランザクション	出カ LEN	出カ ADDR	出カ BURST
INCR パースト					
si.DW [2] = cb.DW [3]	常に成立	1	変更なし	変更なし	INCR
cb.DW [3] = mi.DW [4]	常に成立	1	変更なし	変更なし	INCR
si.DW [2] > cb.DW [3]	if (2*si.SIZE [10] <= cb.Bytes [6])	1	変更なし	変更なし	INCR
	else if (si.downsize_LEN [24] <= 255)	1	si.downsize_LEN [24] - cb.AlignedAdjustment [26]	変更なし	INCR (1)
	else	ceil ((si.downsize_LEN + 1 [24]) / 256)	first = 255 - cb.AlignedAdjustment [26]; last = si.downsize_LEN [24] % 256; others = 255	first = si.ADDR; others = (cb.ADDR[i-1] & ~si.SizeMask [13]) + (256*cb.Bytes [6])	INCR (1)
cb.DW [3] > mi.DW [4]	if (2*cb.SIZE [11] <= mi.Bytes [7])	1	変更なし	変更なし	INCR
	else if (cb.downsize_LEN [25] <= 255)	1	cb.downsize_LEN [25] mi.AlignedAdjustment [27]	変更なし	INCR (1)
	else	ceil ((cb.downsize_LEN + 1 [25]) / 256)	first = 255 - mi.AlignedAdjustment [27]; last = cb.downsize_LEN [25] % 256; others = 255	first = cb.ADDR; others = (mi.ADDR[i-1] & ~cb.SizeMask [14]) + (256*mi.Bytes [24])	INCR (1)
si.DW [2] < cb.DW [3]	if si.CACHE [1]	1	cb.upsize_LEN [18]	変更なし	INCR (1)
	else	1	変更なし	変更なし	INCR
	if (si.CACHE [1])	1	mi.upsize_LEN [21]	変更なし	INCR (1)
	else	1	変更なし	変更なし	INCR

1. 幅変換によってトランザクションの長さが変更される場合、出カ SIZE は常に出力 DATA WIDTH と同じです。

表 2 : 幅変換トランザクション (続き)

相対 DWwidth	条件	出力 トランザクション	出力 LEN	出力 ADDR	出力 BURST
WRAP パースト					
si.DW [2] = cb.DW [3]	常に成立	1	変更なし	変更なし	WRAP
cb.DW [3] = mi.DW [4]	常に成立	1	変更なし	変更なし	WRAP
si.DW [2] > cb.DW [3]	if (2**si.SIZE [10] <= cb.Bytes)	1	変更なし	変更なし	WRAP
	else if (si.downsize_LEN [24] <= 15)	1	si.downsize_LEN [24]	変更なし	WRAP (1)
	else if ((si.ADDR & si.burst_mask [30]) == 0)	1	si.wrap1_LEN [34]	si.ADDR	INCR (1)
	else	2	first = si.wrap1_LEN [34]; second = si.wrap2_LEN [36]	first = si.ADDR; second = si.wrap_address [32]	INCR (1)
cb.DW [3] > mi.DW [4]	if (2**cb.SIZE [11] <= mi.Bytes)	1	変更なし	変更なし	WRAP
	else if (cb.downsize_LEN [25] <= 15)	1	cb.downsize_LEN [25]	変更なし	WRAP (1)
	else if ((cb.ADDR & cb.burst_mask [30]) == 0)	1	cb.wrap1_LEN [35]	cb.ADDR	INCR (1)
	else	2	first = cb.wrap1_LEN; [35] second = cb.wrap2_LEN [37]	first = cb.ADDR; second = cb.wrap_address [33]	INCR (1)
si.DW [2] < cb.DW [3], Write	if (si.CACHE [1])	1	ceil((si.LEN+1) * (2**si.SIZE [10]) / cb.Bytes) - 1	si.wrap_address [32] + (ceil((si.ADDR & si.burst_mask [30]) / cb.Bytes) * cb.Bytes) % si.burst_bytes [28]	If (cb.LEN > 0) then WRAP, else INCR (1)
	else	1	変更なし	変更なし	WRAP
si.DW [2] < cb.DW [3], Read	if (si.CACHE [1])	1	ceil((si.LEN+1) * (2**si.SIZE [10]) / cb.Bytes [6]) - 1	si.wrap_address [33] + (int((si.ADDR & si.burst_mask [30]) / cb.Bytes [6]) * cb.Bytes [6])	If (cb.LEN > 0) then WRAP, else INCR (1)
	else	1	変更なし	変更なし	WRAP

1. 幅変換によってトランザクションの長さが変更される場合、出力 SIZE は常に出力 DATA_WIDTH と同じです。

表 2 : 幅変換トランザクション (続き)

相対 DWidth	条件	出力ランザクション	出力 LEN	出力 ADDR	出力 BURST
cb.DW [3] < mi.DW [4], Write	if (si.CACHE[1])	1	$\text{ceil}((\text{cb.LEN}+1) * (2^{**}\text{cb.SIZE} [11]) / \text{mi.Bytes} [7]) - 1$	$\text{cb.wrap_address} [33] + (\text{ceil}((\text{cb.ADDR} \& \text{cb.burst_mask} [31]) / \text{mi.Bytes} [7]) * \text{mi.Bytes} [7]) \% \text{cb.burst_bytes} [29]$	if (mi.LEN>0) then WRAP, else INCR ⁽¹⁾
	else	1	変更なし	変更なし	WRAP
cb.DW [3] < mi.DW [4], Read	if (si.CACHE[1])	1	$\text{ceil}((\text{cb.LEN}+1) * (2^{**}\text{cb.SIZE} [11]) / \text{mi.Bytes} [7]) - 1$	$\text{cb.wrap_address} [33] + (\text{int}((\text{cb.ADDR} \& \text{cb.burst_mask} [31]) / \text{mi.Bytes} [7]) * \text{mi.Bytes} [7])$	if (mi.LEN>0) then WRAP, else INCR ⁽¹⁾
	else	1	変更なし	変更なし	WRAP
固定長バースト					
si.DW [2] = cb.DW [3]	常に成立	1	変更なし	変更なし	FIXED
cb.DW [3] = mi.DW [4]	常に成立	1	変更なし	変更なし	FIXED
si.DW [2] > cb.DW [3]	if (2**si.SIZE [11] (<= cb.Bytes [6])	1	変更なし	変更なし	FIXED
	else	si.LEN+1	$\text{all} = \max(\text{si.conv_ratio} [22] - \text{cb.AlignedAdjustment} [26] - 1, 0)$	all = si.ADDR	INCR ⁽¹⁾
cb.DW [3] > mi.DW [4]	if (2**cb.SIZE [10] (<= mi.Bytes [7])	1	変更なし	変更なし	FIXED
	else	cb.LEN+1	$\text{all} = \max(\text{cb.conv_ratio} [23] - \text{mi.AlignedAdjustment} [27] - 1, 0)$	all = cb.ADDR	INCR ⁽¹⁾
si.DW [2] < cb.DW [3]	常に成立	1	変更なし	変更なし	FIXED
cb.DW [3] < mi.DW [4]	常に成立	1	変更なし	変更なし	FIXED

1. 幅変換によってトランザクションの長さが変更される場合、出力 SIZE は常に出力 DATA_WIDTH と同じです。

クロック変換

クロック変換は、次の条件に従って実行されます。

- クロックレート リダクション モジュールは、入力 (SI) 側から出力 (MI) 側に向けて整数比 $N:1$ でクロックレートを分周します。
- クロックレート アクセラレーション モジュールは、入力 (SI) 側から出力 (MI) 側に向けて整数比 $1:N$ でクロックレートを逡倍します。
- 同期クロック変換モジュールは、非同期 FIFO を通してチャネル信号を渡すことで、クロックレートを増減します。

リダクションおよびアクセラレーションのいずれのモジュールでも、より高速なクロック ドメインのサンプリング サイクルは自動的に決定されます。各モジュールは 5 つの AXI チャネルすべてに適用されます。

MI および SI はクロック入力のベクターを持ち、各ビットは対応するインターフェイス スロットの全信号と同期しています。AXI インターコネクト コアには専用のネイティブ クロック入力があります。AXI インターコネクト コアは、MI および SI の各スロットのクロック レートをコアのネイティブ クロック レートに自動的に適合させます。

通常、AXI インターコネクト コアのネイティブ クロック入力は、システム デザイン内の最大周波数の SI または MI スロット、たとえばメイン メモリ コントローラーに接続された MI スロットなどが使用しているものと同じクロック ソースに接続されます。

ペリフェラル レジスタ スライス

オプションとして深さ 2 のレジスタ スライス (スキッド バッファ) を、各 SI または MI スロットの 5 つの AXI チャネルのそれぞれに対して挿入し、システムのタイミング クロージャを改善することもできます。SI および MI 両方の最外周部分では、必要に応じて、各インターフェイス スロットの各チャネルにレジスタ スライス バッファを挿入できます。その主な目的は、レイテンシは 1 サイクル分増加しますが、システムのタイミングを改善することです。

ペリフェラル レジスタ スライスは、常に SI または MI スロットのクロックに同期します。

データパス FIFO

一部の状況下では、データ バーストをバッファリングすることで AXI インターコネクトのスループットが向上します。このような状況は通常、データ幅またはクロック レートの変換によって SI または MI スロットのデータ レートが AXI インターコネクト コアのネイティブ データ レートと異なる場合に見られます。レート変換のさまざまな組み合わせに対応するために、オプションで次の箇所にデータ バースト バッファを挿入できます。

- SI 側のデータ幅またはクロック変換の後、クロスバー モジュールの前の SI 側書き込みデータ FIFO
- クロスバー モジュールの後、MI 側のデータ幅、クロックまたはプロトコル変換の前の MI 側書き込みデータ FIFO
- クロスバー モジュールの MI 側の前、MI 側のデータ幅、クロックまたはプロトコル変換の後の MI 側読み出しデータ FIFO
- クロスバー モジュールの SI 側の後、SI 側のデータ幅またはクロック変換の前の SI 側読み出しデータ FIFO

データ FIFO は AXI インターコネクトのネイティブ クロックに同期します。各データ FIFO の幅は、AXI インターコネクトのネイティブ データ幅と同じです。詳細は 49 ページの「データパス FIFO」を参照してください。

ID 信号の使用方法

SI から MI へ伝搬するトランザクション ID 信号 (AWID と ARID) および MI から SI へ伝搬するトランザクション ID 信号 (BID と RID) によって、各トランザクションの開始元と、MI で受信した応答が元の SI スロットに戻される経路が、システムのインターコネクト トポロジ上で特定されます。

エンドポイントのマスター デバイスは、トランザクションの複数の「スレッド」を選択するのに使用できる AWID 信号と ARID 信号を必要に応じて出力し、マスター IP が内部的に複数のマスター デバイスで構成されているかのように動作します。「順序変更の深さ」は 1 つのマスターが生成可能な ID 値の合計であり、 $2^{idwidth}$ で表されます。ここで、idwidth は各 SI スロットの THREAD_ID_WIDTH パラメーターで指定されます。順序変更の深さが 1 のマスター デバイスは、インターフェイスに ID 信号は必要がありません。トランザクションの順序付けは次のように決定されます。

- 同一スレッドに属するトランザクションは、受付順に応答を返す必要がある。
- 異なるスレッドのトランザクションは、アウトオブオーダーで応答を返すことができる。

すべての SI スロット ID 値は、MI スロットのいずれかに伝搬される前の時点で一意でなければなりません。AXI インターコネクト コアは、各 SI スロットでサンプリングされる AWID 信号および ARID 信号 (存在する場合) の前に、一意の定数「マスター ID」値を付加します。

各 SI スロットに対応付けられた BASE_ID パラメーターにより、AXI インターコネクト コアはコンパイル時にマスター ID を割り当てることができます。エンドポイントのマスター デバイスは、割り当てられたマスター ID を自身の ID 出力に入れる必要はないため、割り当てられたマスター ID 値を認識する必要はありません。

2 つのインターコネクト インスタンスがカスケード接続され、一方のインスタンスの MI スロットが他方のインスタンスの SI スロットに接続されている場合、アップストリームの AXI インターコネクト コアで生成されたすべての ID 信号が、接続されたマスター デバイスのスレッド ID ビットであるかのように扱われます。ほかのマスター デバイスと同様に、ダウンストリームの AXI インターコネクト コアはカスケード接続された SI スロットからサンプリングされた ID 信号の前に一意のマスター ID を付加します。これにより、カスケード接続された AXI インターコネクト トポロジを伝搬するにつれて ID 幅が大きくなります。マスター ID が一致する応答すべてがアップストリームの AXI インターコネクト インスタンスに戻されます。

M_AXI_WID 信号は、接続された AXI3 スレーブ デバイスのすべてに与えられます。この信号は通常、対応する AW チャネル転送に対して発行された M_AXI_AWID の値に基づいて生成されます。S_AXI_WID 信号は、インターコネクトがパススルーモード (SI : 1 スロット、MI : 1 スロット) に設定され、接続されたマスターとスレーブ デバイスがいずれも AXI3 の場合 (かつ、ほかの変換を実行しない場合) を除き、通常無視されます。その場合の S_AXI_WID 信号は、ほかのすべての AXI インターフェイス信号と共に直接 M_AXI_WID に伝搬されます。

図 10 に、2 つの AXI インターコネクト インスタンスをカスケード接続した場合の例を示します。

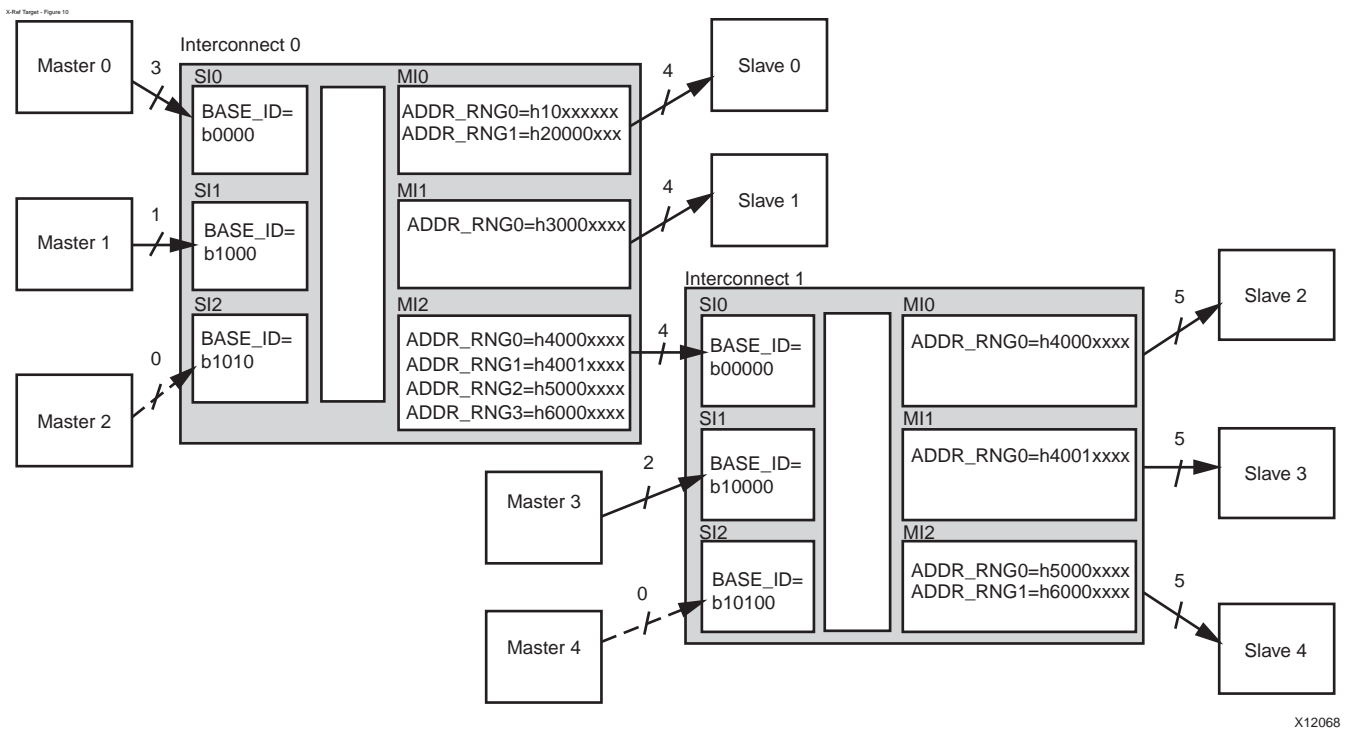


図 10 : AXI インターコネクト コアのカスケード接続

図 10 は、次のことを示しています。

- AXI インターコネクト 0 の MI スロット 2 (MI2) は、AXI インターコネクト 1 の SI スロット 0 (SI0) に接続されています。エンドポイント スレーブ デバイス 2 ~ 4 は、AXI インターコネクト 1 の MI0 ~ MI2 で定義されたアドレス範囲を持ちます。
注記：わかりやすくするために、BASEADDR と HIGHADDR のペアを don't care を含む ADDR 範囲として表しています。
- インターコネクト 1 がアクセスできるアドレス範囲のすべてが、インターコネクト 0 の MI2 の複数のアドレス範囲として列挙されています。
- 矢印は、各マスター デバイスから伝搬する ID 信号を表します。AXI インターコネクト 0 は 4 ビットの ID 出力を生成します。これは、マスター ID の一意性を確保するために必要な最小の幅です。たとえば、マスター 0 がトランザクションを発行する場合の出力 ID は、マスター ID (1'b0) の後にマスター デバイスからサンプリングされた 3 ビットの ID を付加したのになります。
- マスター 2 からのトランザクションはすべて、ID 値 4'b1010 を持ちます (マスター デバイスから可変スレッド ビットは出力されない)。
- マスター 0 ~ 2 からのトランザクションの宛先がスレーブ 2 ~ 4 の場合、AXI インターコネクト 0 は 4 ビットの ID 値をインターコネクト 1 に渡します。次に、インターコネクト 1 がこの値の前に 1'b0 (SI0 のマスター ID) を付加して 5 ビットの ID を生成し、接続されているスレーブ デバイスのいずれかに渡します。

複数アドレス範囲のサポート

AXI インターコネクト コアは SI スロットからの各 AW および AR チャネル トランザクションのアドレスをデコードして、どの MI スロットが各トランザクションの宛先なのかを判断する必要があります。このアドレス デコードには、MI スロットの識別に必要な上位アドレス ビットのみが関係し、接続されたスレーブ デバイス内での位置を識別するために使用する下位ビットは無視されます。SI から受信したアドレス値全体が MI に渡されてスレーブ デバイスで使用できます。上位のアドレス ビットが通常スレーブ デバイスで再利用されない場合でも、接続したモニターには表示できます。

1 つのスレーブ デバイス (MI スロット) にアクセスするときに、連続していない、複数のアドレス範囲を定義する場合もあります。AXI インターコネクト コアのアドレス デコード ロジックには、各 MI スロットの選択を決定する複数のアドレス範囲が含まれます。複数のアドレス範囲の識別は、一般に接続されたスレーブ デバイスの機能でも必要です。

これは、AXI インターコネクト コアによってインプリメントされているデコード ロジックの一部がスレーブ デバイスにも複製されていることを意味します。AMBA 4 仕様では、AXI インターコネクト コアによってデコードされるアドレス範囲をエンコードするときに使用する AXI 信号 `AWREGION` と `ARREGION` が導入されました。AXI インターコネクト コアが生成したこのような `REGION` 出力を、複数のアドレス デコード範囲を持つスレーブ デバイスが使用することで、スレーブ デバイスでアドレス範囲デコード ロジックを重複して持つことがなくなります。

各 `REGION` 信号で生成される 4 ビットの値は各 MI スロット内の `C_M_AXI_BASE_ADDR` および `C_M_AXI_HIGH_ADDR` パラメータのトランザクション アドレスが一致する位置に対応します。多くの場合、これらのアドレス範囲は接続されたスレーブ デバイスの複数のパラメータを用いて、`C_busif_RNGnn_BASEADDR` や `C_busif_RNGnn_HIGHADDR` のような形式で表されます。さまざまな MI スロットに複数のアドレス範囲を割り当てる方法については、図 10 の例を参照してください。

SI で受信したトランザクション アドレスが AXI インターコネクト コアによってデコードされたアドレス範囲のいずれかにない場合、トランザクションはトラップされ、AXI インターコネクト コア内のデコード エラー モジュールで処理されます。AXI インターコネクト コアの MI スロットが 1 つのみで、アドレス範囲が 1 つしかない場合は、例外が発生します。この場合、`C_RANGE_CHECK` パラメータによって、アドレスのデコードおよび関連するデコード エラー トラップがインプリメントされているかどうか、またはすべてのトランザクションを MI スロットに伝搬するかどうかが判別されます。

サイクル依存性の回避

Multiple outstanding トランザクションを発行できる (1 つまたは複数のマスター デバイスによって発行された) 2 つ以上のトランザクション ID が存在し、キューに複数トランザクションを保持できる 2 つ以上のスレーブ デバイスが接続されており、スレーブ デバイスのいずれかが R チャネルまたは B チャネルにアウトオブオーダーで応答できる場合、サイクル依存性 (デッドロック) のリスクがあります。AXI 完全準拠であるため、AXI インターコネクト コアにはアウトオブオーダー応答をサポートするスレーブ デバイスの処理が組み込まれています。

デッドロックの起きる仕組み

次の例では、読み出しトランザクションがデッドロックになるシーケンスを示します。スレーブ デバイス書き込み応答の順序を変更できる場合、同様の状況が書き込みトランザクションのシーケンスにも当てはまります。この例は、2 つのマスター デバイス (M0 と M1)、および 2 つのスレーブ デバイス (S0 と S1) が AXI インターコネクト コアで接続されている場合を示しています。

1. マスター デバイス M0 がスレーブ デバイス S0 からの読み出しを行います。
2. 次に、マスター デバイス M0 がスレーブ デバイス S1 からの読み出しを (同じ ID のスレッドを使用して) 行います。
3. 続いて、マスター デバイス M1 がスレーブ デバイス S1 からの読み出しを行います。
4. その後、マスター デバイス M1 がスレーブ デバイス S0 からの読み出しを (同じ ID のスレッドを使用して) 行います。
5. スレーブ デバイス S0 は最初にマスター デバイス M1 に応答します。読み出し応答の順序が変更されていますが、これは受信したトランザクション ID が異なるため許容されます。ただし、マスター デバイス M1 は先にスレーブ デバイス S1 からの応答を受け取る必要があるため、AXI インターコネクト コアは応答をマスター デバイス M1 に渡すことができません。
6. スレーブ デバイス S1 がマスター デバイス M0 (順序変更されていない) に応答します。しかし、マスター デバイス M0 は先にスレーブ デバイス S0 からの応答を受け取る必要があるため、AXI インターコネクト コアは応答をマスター デバイス M0 に渡すことができません。

これにより、デッドロックが生じます。

1 スレーブ 1 ID でデッドロックを回避

AXI インターコネクト コアでデッドロックを回避するには、「1 スレーブ 1 ID」方式を使用します。この方式は、最も重要な問題であるトランザクションのパフォーマンスに影響を与えません。これは、複数の読み出しおよび書き込みのパイプライン処理であり、メモリ コントローラーなどのパフォーマンスに厳しいスレーブ デバイスに接続した複数のマスター デバイスで使われます。

「1 スレーブ 1 ID」方式を採用すると、各 SI スロットで受信される (各マスター デバイスからの) ID スレッドが持つことのできる (各種の) Outstanding トランザクションは、一度に 1 つの MI スロットのみに制限されます。

ただし、MI スロットが、複数の SI スロットからの Multiple Outstanding トランザクションを発行できることに変わりはありません。

前セクションに示した例にこの規則を適用すると、手順 2 の M0 から S1 への読み出しトランザクションは、S0 が M0 への応答を完了するまで待たされます。同様に、手順 4 の M1 から S0 へのトランザクションは S1 が M1 への応答を完了するまで待たされます。これらの条件のもとでは、どのようにトランザクションが処理されようとデッドロックの原因となる相互依存は回避されます。

「1 スレーブ 1 ID」による制約は、AXI インターコネクト コアが 1 対 1 のシンプルなパススルーで設定されている場合を除いて、すべてのトランザクション スレッドに適用されます。デッドロックを回避するほか、この制約により、連続したトランザクションのうちの 1 つのトランザクション スレッドが別の MI スロットを宛先に行っていた場合でも、宛先 SI でのすべての書き込みトランザクションが順序どおり完了することが保証されます。

たとえば、マスター デバイスがメモリの DMA ディスクリプターに書き込みを行った後に、そのディスクリプターを読み出した DMA エンジンの制御レジスタに書き込みを行います。AXI インターコネクト コアは最初の書き込みが完了する (メモリ コントローラーから書き込み応答を受信する) までは 2 回目の書き込みを DMA スレーブ デバイスに伝搬させないため、DMA がメモリから古いディスクリプターのデータを読み出すことはありません。したがって各マスター デバイスでは、さまざまなスレーブ デバイス、同じ方向、同じスレッドのトランザクションが順序どおり完了することが保証されます。これにより、このような条件のもとでは、先行トランザクションの書き込み応答を受信してから後続の書き込みトランザクションを発行するという条件をマスター デバイスに設定する必要がなくなります。

注記：AXI プロトコルでは、前の書き込みがすべて完了したという B チャネル応答を待つ以外に、書き込みトランザクションと読み出しトランザクションの間で順序どおりの完了を保証する手段が用意されていません。

エラー信号

AXI インターコネクト コアで検出されるエラー状態は次のとおりです。

- アドレス デコード エラー：コネクティビティ マップと適用可能な書き込み専用/読み出し専用パラメーターに基づくトランザクションのアドレスに、使用可能な MI スロットがマッピングされていません。AXI インターコネクト コアが DECERR を返し、トランザクションはどの MI スロットにも伝搬されません。ただし、アドレス デコード エラーは、C_RANGE_CHECK パラメーターが 0 に設定されている場合にはトラップされません。デフォルトでは、複数の MI スロットが存在するか、複数のアドレス範囲がある場合は、常に C_RANGE_CHECK がイネーブルになります。複数の MI スロットがあり、C_RANGE_CHECK パラメーターが強制的に OFF (0) に設定されている場合、無効なアドレスにアクセスすると、プロトコルに準拠しない予測不能なトランザクションの伝搬が発生する可能性があります。
- AXI4-Lite アクセス違反：次の条件のいずれかで発生します。
 - バースト長違反：AXI4-Lite MI スロットが宛先であるときに、トランザクションの長さが 1 データ ビートよりも大きい。
 - データ サイズ違反：AXI4-Lite MI スロットが宛先であるときに、トランザクション データの転送サイズの幅が 4 バイトよりも大きい。

AXI インターコネクト コアが DECERR を返し、トランザクションは MI スロットに伝搬されません。AXI4-Lite のアクセス違反は、C_RANGE_CHECK = 0 の場合に無効になります。デフォルトでは、いずれかの MI スロットが AXI4-Lite として構成され、いずれかの SI スロットが AXI4-Lite 以外のプロトコルとして構成された場合、C_RANGE_CHECK がイネーブルになります。C_RANGE_CHECK が OFF (0) のときに、AXI4-Lite MI スロットを宛先とする無効なトランザクションが発行されると、結果が予測不能となり、トランザクションがエラーになる可能性が高くなります。

- C_M_AXI_SECURE を設定された MI スロットが、AWPROT[1] または ARPROT[1] をセットされた (セキュアでない) トランザクションの宛先となっています。

注記: MI スロットのいずれかが SECURE として設定されている場合には、C_RANGE_CHECK を無効にできません。

AXI インターコネクト コアは次のエラー状態を検出しません。

- MI スロットで受信された応答 ID がいずれの SI スロットにもマッピングされない場合、AXI インターコネクト コアからの READY 応答は MI スロットで発行されません。応答全体 (書き込み応答または読み出しデータ バースト) は AXI インターコネクト コアにより永続的に遮断されます。これにより、問題のあるスレーブ デバイスと応答の受信を待つマスター デバイスがハングアップする場合があります。
- AXI インターコネクト コアは AXI4 プロトコル違反をトラップしません。これはエンドポイント IP で行われます。
- AXI インターコネクト コアは書き込みデータのインターリーブをサポートせず、トラップすることはありません (すべての書き込みデータは書き込みトランザクションの順序に従って振り分け。WID は SI でサンプリングされない)。
- AXI インターコネクト コアは幅の狭いバースト違反をトラップしません。この違反は、SI スロットが C_S_AXI_SUPPORTS_NARROW_BURST = 0 に設定されている場合に、長さが 1 データ ビートを超え、データ転送サイズが SI スロットのデータ幅よりも狭いトランザクションを受信するか、AWCACHE[1] または ARCACHE[1] がディアサートされたトランザクションを受信すると発生します。これは、エンドポイントのマスター IP で行われます。
- Xilinx Platform Studio (XPS) は、コンパイル時にエラーとなるコンフィギュレーションを行わないようにするデザインルールを適用します。したがって、AXI インターコネクト コアには次のコンフィギュレーション エラーに対するエラー検出ロジックは備えられていません。
 - 非同期クロック コンフィギュレーション以外でクロック倍率が整数でない
 - パラメーター値範囲違反
 - アドレスまたは ID の範囲が重複している、非バイナリ サイズである、またはベース値のアライメントが正しくない

AXI プロトコル コンバーター

次のサブセクションで、AXI プロトコル コンバーターについて説明します。

- [AXI4-Lite スレーブ変換](#)
- [AXI3 スレーブ コンバーター](#)

AXI4-Lite スレーブ変換

AXI4-Lite スレーブ デバイスに接続された各 MI スロットは、AXI4-Lite 変換ブロックを通してルーティングされます。変換ブロックでは、書き込みトランザクションと読み出しトランザクション間の単一スレッドのラウンドロビン アービトレーションを含めて、すべてのトランザクションを単一スレッド化します。多くの場合、書き込みアドレスと読み出しアドレスは 1 つのバスに多重化され、その後 MI スロットの AWADDR 信号と ARADDR 信号に複製されます。多くの場合、これらの重複する信号はバックエンド デザインのインプリメンテーション時に自動削除されるため、AXI4-Lite のスレーブが使用するリソースはアドレスバスが 1 つしかない場合とほぼ同様になります。

トランザクション ID (AWID または ARID) は分解されて変換ブロックに格納され、応答の転送時に BID または RID として読み出されます。

注記：AXI4-Lite ソリューションのリソース使用率を最小限に抑えるため、AXI4/AXI3 パーストが AXI4-Lite プロトコル変換によって AXI4-Lite スレーブ向けの一連のシングル ビート トランザクションに変換されることはありません。AXI4-Lite スレーブ デバイスにアクセスするマスター デバイスは、データ転送の SIZE が 32 ビットを超えないシングル ビート トランザクションしか発行しないものと見なされます。

図 11 に AXI4-Lite の変換ロジックを示します。

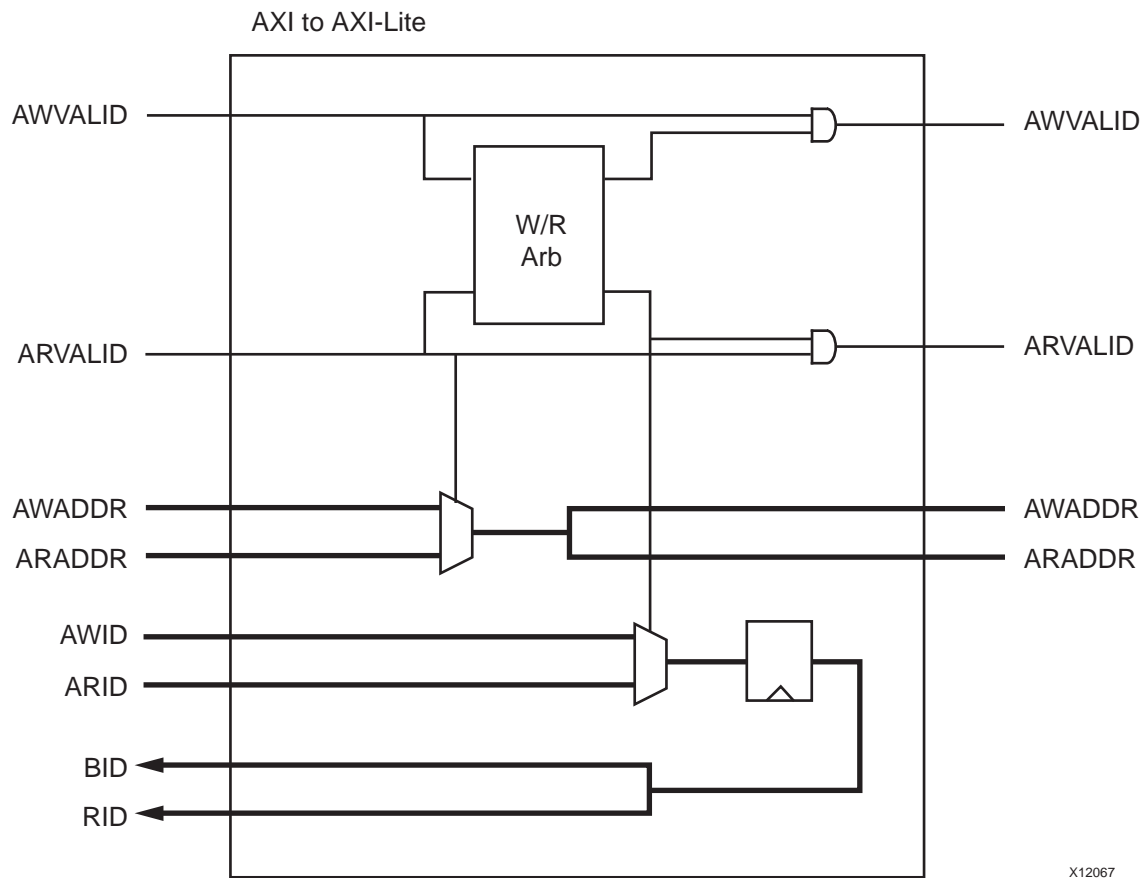


図 11 : AXI4-Lite の変換ロジック

X12067

AXI3 スレーブ コンバーター

MI スロットが 1 つ以上の AXI4 SI スロットからアクセスできる場合、AXI3 スレーブ変換モジュールは AXI3 スレーブ デバイスに接続された各 MI スロットの位置にインスタンス化されます。

アドレス チャンネル ダウンサイザー モジュールと同様に、このモジュールはスレーブ インターフェイスで AW または AR 転送 (コマンド) を受信し、MI で 1 つまたは複数のコマンドを生成します。データ転送の SIZE が AXI3 コンバーターで変更されることはありません。16 データ ビートよりも長いバーストを受信した場合、コマンドは複数の短いバースト トランザクションに分割されます。

AXI3 コンバーター モジュールは通常、複数の Outstanding トランザクションを伝搬できます。AW/AR チャンネル転送からの トランザクション特性として、対応する応答転送を待機する間、キューに入ります。ただし、書き込み応答または読み出しデータの順序が変更する可能性があるため、AW および AR チャンネル コンバーターによる トランザクションの受け入れは、トランザクションが分割を必要としている場合は常に、各方向一度に 1 つの トランザクションに限られます。

I/O 信号

このセクションでは、AXI インターコネクト コア信号の一覧を示します。

表 3、30 ページの表 4、32 ページの表 5、33 ページの表 6、34 ページの表 7、37 ページの表 8 の「デフォルト」列には、入力信号が必須 (REQ) かどうか、必須でない場合は未接続の場合のデフォルト値を示しています。信号の接続が必要なのは、使用する SI スロットおよび MI スロットのみです。「デフォルト」列には、AXI4、AXI3、Lite (AXI4-Lite) など、スロットのプロトコル モードを示す値も記載されています。AXI4-Lite でサンプリングされない (ドントケア) 入力信号は「d/c」で示します。

スレーブ インターフェイスの I/O 信号

表 3 に、スレーブ インターフェイス信号の一覧を示します。「幅」列の「N」は、SI スロットの総数を示します。これは、AXI インターコネクト コアに接続されたマスター デバイスの数です。

CORE Generator ツール フローを使用した場合、表 3 に記載された各信号名は、Snn_AXI_signalname の形式となります。ここで、nn は、各スレーブ インターフェイスの 2 桁のインデックス番号 (先行ゼロ付き) です。CORE Generator コア インターフェイスでは、「幅」列はすべての信号で N = 1 となります。

表 3 : スレーブの I/O 信号

信号名	方向	デフォルト	幅	説明 (範囲)
S_AXI_ARESET_OUT_N	出力		N*1	出力 (アクティブ Low) をリセットし、各スロットのクロックに再同期する (AXI プロトコルで定義された信号ではない)
S_AXI_ACLK	入力	REQ	N*1	クロック
S_AXI_AWID	入力	AXI3、AXI4 : 0 Lite : d/c	N*C_AXI_ID_WIDTH	書き込みアドレス チャンネル トランザクション ID
S_AXI_AWADDR	入力	REQ	N*C_AXI_ADDR_WIDTH	書き込みアドレス チャンネルのアドレス
S_AXI_AWLEN	入力	AXI3、AXI4 : 0 Lite : d/c	N*8	書き込みアドレス チャンネルのバースト長 (0 ~ 255)
S_AXI_AWSIZE	入力	AXI3、AXI4 : REQ ⁽¹⁾ Lite : d/c	N*3	書き込みアドレス チャンネルの転送サイズコード (0 ~ 7)
S_AXI_AWBURST	入力	AXI3、AXI4 : REQ ⁽¹⁾ Lite : d/c	N*2	書き込みアドレス チャンネルのバースト タイプコード (0 ~ 2)
S_AXI_AWLOCK	入力	AXI3、AXI4 : 0 Lite : d/c	N*2	書き込みアドレス チャンネルのアトミック アクセスタイプ (0、1)

表 3 : スレーブの I/O 信号 (続き)

信号名	方向	デフォルト	幅	説明 (範囲)
S_AXI_AWCACHE	入力	AXI3, AXI4 : 0 ⁽²⁾ Lite : d/c	N*4	書き込みアドレス チャンネル キャッシュ 特性
S_AXI_AWPROT	入力	0b000 ⁽³⁾	N*3	書き込みアドレス チャンネルの保護ビット
S_AXI_AWQOS ⁽⁴⁾	入力	AXI4 : 0 Lite : d/c	N*4	AXI4 書き込みアドレス チャンネルのサービス品質 (QoS)
S_AXI_AWUSER ⁽⁵⁾	入力	AXI3, AXI4 : 0 Lite : d/c	N*C_AXI_AWUSER_WIDTH	ユーザー定義の AW チャンネル信号
S_AXI_AWVALID	入力	REQ	N*1	書き込みアドレス チャンネル有効
S_AXI_AWREADY	出力		N*1	書き込みアドレス チャンネル準備完了
S_AXI_WID ⁽⁵⁾	入力	0	N*C_AXI_ID_WIDTH	AXI3 マスターの書き込みデータ チャンネル トランザクション ID (インターコネクトが AXI3 エンドポイント間のパスト スループ モードに設定されている場合に、サンプリングされ、伝搬される)
S_AXI_WDATA	入力	REQ	N*C_S_AXI_DATA_WIDTH	書き込みデータ チャンネルのデータ
S_AXI_WSTRB	入力	すべて 1	N*C_S_AXI_DATA_WIDTH/8	書き込みデータ チャンネルのバイト ストロープ
S_AXI_WLAST	入力	AXI3, AXI4 : 0 Lite : d/c	N*1	書き込みデータ チャンネルの最終データ ビット
S_AXI_WUSER ⁽⁵⁾	入力	AXI3, AXI4 : 0 Lite : d/c	N*C_AXI_WUSER_WIDTH	ユーザー定義の W チャンネル信号
S_AXI_WVALID	入力	REQ	N*1	書き込みデータ チャンネル有効
S_AXI_WREADY	出力		N*1	書き込みデータ チャンネル準備完了
S_AXI_BID	出力		N*C_AXI_ID_WIDTH	書き込み応答 チャンネル トランザクション ID
S_AXI_BRESP	出力		N*2	書き込み応答チャンネルの応答コード (0 ~ 3)
S_AXI_BUSER ⁽⁵⁾	出力		N*C_AXI_BUSER_WIDTH	ユーザー定義の B チャンネル信号
S_AXI_BVALID	出力		N*1	書き込み応答チャンネル有効
S_AXI_BREADY	入力	REQ	N*1	書き込み応答チャンネル準備完了
S_AXI_ARID	入力	AXI3, AXI4 : 0 Lite : d/c	N*C_AXI_ID_WIDTH	読み出しアドレス チャンネル トランザクション ID
S_AXI_ARADDR	入力	REQ	N*C_AXI_ADDR_WIDTH	読み出しアドレス チャンネルのアドレス
S_AXI_ARLEN	入力	AXI3, AXI4 : 0 Lite : d/c	N*8	読み出しアドレス チャンネルのバースト長コード (0 ~ 255)
S_AXI_ARSIZE	入力	AXI3, AXI4 : REQ ⁽¹⁾ Lite : d/c	N*3	読み出しアドレス チャンネルの転送サイズコード (0 ~ 7)
S_AXI_ARBURST	入力	AXI3, AXI4 : REQ ⁽¹⁾ Lite : d/c	N*2	読み出しアドレス チャンネルのバースト タイプ (0 ~ 2)
S_AXI_ARLOCK	入力	AXI3, AXI4 : 0 Lite : d/c	N*2	読み出しアドレス チャンネルのアトミック アクセス タイプ (0, 1)
S_AXI_ARCACHE	入力	AXI3, AXI4 : 0 ⁽²⁾ Lite : d/c	N*4	読み出しアドレス チャンネル キャッシュ 特性
S_AXI_ARPROT	入力	0b000 ⁽³⁾	N*3	読み出しアドレス チャンネルの保護ビット

表 3: スレーブの I/O 信号 (続き)

信号名	方向	デフォルト	幅	説明 (範囲)
S_AXI_ARQOS ⁽⁴⁾	入力	AXI4 : 0 Lite : d/c	N*4	AXI4 読み出しアドレス チャンルのサービス品質 (QoS)
S_AXI_ARUSER ⁽⁵⁾	入力	AXI3, AXI4 : 0 Lite : d/c	N*C_AXI_ARUSER_WIDTH	ユーザー定義の AR チャンネル信号
S_AXI_ARVALID	入力	REQ	N*1	読み出しアドレス チャンネル有効
S_AXI_ARREADY	出力		N*1	読み出しアドレス チャンネル準備完了
S_AXI_RID	出力		N*C_AXI_ID_WIDTH	読み出しデータ チャンネル トランザクション ID
S_AXI_RDATA	出力		N*C_S_AXI_DATA_WIDTH	読み出しデータ チャンネルのデータ
S_AXI_RRESP	出力		N*2	読み出しデータ チャンネル応答コード (0 ~ 3)
S_AXI_RLAST	出力		N*1	読み出しデータ チャンネルの最終データ ビート
S_AXI_RUSER ⁽⁵⁾	出力		N*C_AXI_RUSER_WIDTH	ユーザー定義の R チャンネル信号
S_AXI_RVALID	出力		N*1	読み出しデータ チャンネル有効
S_AXI_RREADY	入力	REQ	N*1	読み出しデータ チャンネル準備完了

- AXI4 マスター デバイスの AW/RSIZE 出力および AW/RBURST 出力は、マスター デバイス自身で駆動することを推奨します。通常、アプリケーション要件に規定されていない限り、マスター デバイスはそのインターフェイスのデータ幅に対応した AW/RSIZE の値を駆動します。通常マスター デバイスはその AW/RBURST 出力を 0b01 に駆動します。これは、インクリメンタル (INCR) パーストを表します。
- マスター デバイスがそれ自身の AW/RCACHE 出力を 0b0011 に駆動し、幅変換の実行中に AXI インターコネク コアがデータを圧縮できるようにして、データバス FIFO でのストア フォワードを可能にすることを推奨します。
- AXI プロトコルの場合、マスター デバイスは自身の AW/RPROT 出力を駆動する必要があります。AW/RPROT 信号が駆動されないままになっている場合、デフォルトですべて 0 となり、トランザクションはセキュアであると解釈されます。
- QOS は AXI4 プロトコル仕様でのみ定義されている信号ですが、このインターコネク IP は AXI3 として設定された SI スロットに対しても QOS 信号を伝搬します。
- CORE Generator ツール フローを使用した場合は、適用されません。

マスター インターフェイスの I/O 信号

表 4 で「幅」列の「M」は、マスター インターフェイス (MI) スロットの総数を示します。これは、AXI インターコネク コアに接続されたスレーブ デバイスの数です。

CORE Generator ツール フローを使用した場合、表 4 に記載された各信号名は、Mmm_AXI_signalname の形式となります。ここで、mm は現時点では常に 00 です。

表 4: マスター I/O 信号

信号名	方向	デフォルト	幅	説明 (範囲)
M_AXI_ARESET_OUT_N	出力		M*1	出力 (アクティブ Low) をリセットし、各スロットのクロックに再同期する (AXI プロトコルで定義された信号ではない)
M_AXI_ACLK	入力	REQ	M*1	クロック
M_AXI_AWID	出力		M*C_AXI_ID_WIDTH	書き込みアドレス チャンネル トランザクション ID
M_AXI_AWADDR	出力		M*C_AXI_ADDR_WIDTH	書き込みアドレス チャンネルのアドレス
M_AXI_AWLEN	出力		M*8	書き込みアドレス チャンネルのバースト長コード (0 ~ 255)
M_AXI_AWSIZE	出力		M*3	書き込みアドレス チャンネルの転送サイズ コード (0 ~ 7)
M_AXI_AWBURST	出力		M*2	書き込みアドレス チャンネルのバースト タイプ (0 ~ 2)
M_AXI_AWLOCK	出力		M*2	書き込みアドレス チャンネルのアトミック アクセス タイプ (0, 1)
M_AXI_AWCACHE	出力		M*4	書き込みアドレス チャンネル キャッシュ 特性
M_AXI_AWPROT	出力		M*3	書き込みアドレス チャンネルの保護ビット

表 4: マスター I/O 信号 (続き)

信号名	方向	デフォルト	幅	説明 (範囲)
M_AXI_AWREGION ⁽¹⁾	出力		M*4	AXI4 書き込みアドレス チャンネルのアドレス領域インデックス
M_AXI_AWQOS ⁽²⁾	出力		M*4	書き込みアドレス チャンネルのサービス品質 (QoS)
M_AXI_AWUSER ⁽¹⁾	出力		M*C_AXI_AWUSER_WIDTH	ユーザー定義の AW チャンネル信号
M_AXI_AWVALID	出力		M*1	書き込みアドレス チャンネル有効
M_AXI_AWREADY	入力	REQ	M*1	書き込みアドレス チャンネル準備完了
M_AXI_WID ⁽¹⁾	出力		M*C_AXI_ID_WIDTH	AXI3 スレーブの書き込みデータ チャンネル トランザクション ID (通常、M_AXI_AWID からコピー)
M_AXI_WDATA	出力		M*C_M_AXI_DATA_WIDTH	書き込みデータ チャンネルのデータ
M_AXI_WSTRB	出力		M*C_M_AXI_DATA_WIDTH/8	書き込みデータ チャンネルのデータ バイト ストロップ
M_AXI_WLAST	出力		1	書き込みデータ チャンネルの最終データ ビート
M_AXI_WUSER ⁽¹⁾	出力		M*C_AXI_WUSER_WIDTH	ユーザー定義の W チャンネル信号
M_AXI_WVALID	出力		M*1	書き込みデータ チャンネル有効
M_AXI_WREADY	入力	REQ	M*1	書き込みデータ チャンネル準備完了
M_AXI_BID	入力	AXI3, AXI4: REQ Lite: d/c	M*C_AXI_ID_WIDTH	書き込み応答 チャンネル トランザクション ID
M_AXI_BRESP	入力	0b00	M*2	書き込み応答チャンネルの応答コード (0 ~ 3)
M_AXI_BUSER ⁽¹⁾	入力	AXI3, AXI4: 0 Lite: d/c	M*C_AXI_BUSER_WIDTH	ユーザー定義の B チャンネル信号
M_AXI_BVALID	入力	REQ	M*1	書き込み応答チャンネル有効
M_AXI_BREADY	出力		M*1	書き込み応答チャンネル準備完了
M_AXI_ARID	出力		M*C_AXI_ID_WIDTH	読み出しアドレス チャンネル トランザクション ID
M_AXI_ARADDR	出力		M*C_AXI_ADDR_WIDTH	読み出しアドレス チャンネルのアドレス
M_AXI_ARLEN	出力		M*8	読み出しアドレス チャンネルのバースト長コード (0 ~ 255)
M_AXI_ARSIZE	出力		M*3	読み出しアドレス チャンネルの転送サイズ コード (0 ~ 7)
M_AXI_ARBURST	出力		M*2	読み出しアドレス チャンネルのバースト タイプ (0 ~ 2)
M_AXI_ARLOCK	出力		M*2	読み出しアドレス チャンネルのアトミック アクセス タイプ (0、1)
M_AXI_ARCACHE	出力		M*4	読み出しアドレス チャンネル キャッシュ特性
M_AXI_ARPROT	出力		M*3	読み出しアドレス チャンネルの保護ビット
M_AXI_ARREGION ⁽¹⁾	出力		M*4	AXI4 読み出しアドレス チャンネルのアドレス領域インデックス
M_AXI_ARQOS ⁽²⁾	出力		M*4	AXI4 読み出しアドレス チャンネルのサービス品質 (QoS)
M_AXI_ARUSER ⁽¹⁾	出力		M*C_AXI_ARUSER_WIDTH	ユーザー定義の AR チャンネル信号
M_AXI_ARVALID	出力		M*1	読み出しアドレス チャンネル有効
M_AXI_ARREADY	入力	REQ	M*1	読み出しアドレス チャンネル準備完了
M_AXI_RID	入力	AXI3, AXI4: REQ Lite: d/c	M*C_AXI_ID_WIDTH	読み出しデータ チャンネル トランザクション ID
M_AXI_RDATA	入力	REQ	M*C_M_AXI_DATA_WIDTH	読み出しデータ チャンネルのデータ
M_AXI_RRESP	入力	0b00	M*2	読み出しデータ チャンネルの応答コード (0 ~ 3)

表 4: マスター I/O 信号 (続き)

信号名	方向	デフォルト	幅	説明 (範囲)
M_AXI_RLAST	入力	AXI3, AXI4: REQ Lite: d/c	M*1	読み出しデータ チャンネルの最終データ ビート
M_AXI_RUSER ⁽¹⁾	入力	AXI3, AXI4: 0 Lite: d/c	M*C_AXI_RUSER_WIDTH	ユーザー定義の R チャンネル信号
M_AXI_RVALID	入力	REQ	M*1	読み出しデータ チャンネル有効
M_AXI_RREADY	出力		M*1	読み出しデータ チャンネル準備完了

1. CORE Generator ツールフローを使用した場合は、適用されません。
2. QOS は AXI4 プロトコル仕様でのみ定義されている信号ですが、このインターコネクト IP は AXI3 として設定された MI スロットに対しても QOS 信号を伝搬します。

グローバル ポー

表 5: グローバル ポート信号

ポート信号名	方向	デフォルト	幅	説明 (範囲)
INTERCONNECT_ACLK	入力	REQ	1	インターコネクトのネイティブ クロック入力
INTERCONNECT_ARESETN	入力	REQ	1	グローバル リセット (アクティブ Low)。詳細は「 リセット要件 」を参照。

リセット要件

すべての内部ロジックのリセットが確実に完了するように、INTERCONNECT_ARESETN 入力を 16 クロック サイクル間以上アクティブ (Low) に保持する必要があります。複数のクロック周波数を使用している場合は、AXI インターコネクト コアに接続されている最低周波数 (INTERCONNECT_ACLK の周波数も含む) のクロックの 16 サイクル間 INTERCONNECT_ARESETN をアクティブにする必要があります。この要件は、INTERCONNECT_ARESETN を proc_sys_reset コアの同様の名前の出力ポートによって駆動することで満たすことができます。

デザイン パラメーター

次のサブセクションに、デザイン パラメーターとこれらのパラメーターの説明に使用する表記規則の一覧を示します。

パラメーター サマリ テーブルにおける表記規則

コア、スレーブ固有、マスター固有のパラメーター テーブル (表 6、表 7、および 表 8) では、次の表記規則を使用しています。

「表示形式/範囲」列は次のように表記されます。

- 「N」は C_NUM_SLAVE_SLOTS の値を表します。
- 「M」は C_NUM_MASTER_SLOTS の値を表します。
- 中かっこ { } は後に続く値の繰り返し回数を示します。
- 「Bit1」は 1 ビットの値を、「Bit32」は 32 ビットの値を、「Bit64」は 64 ビットの値を表します。
たとえば、「{N} Bit32」は 32 ビットの値が SI スロットごとに繰り返されるパラメーターを表します。
- 脚注 N が付いていない限り、コアのパラメーターは HDL のコンパイルに影響を与えます。

グローバル コア パラメーター (XPS フロー)

表 6: グローバル コア パラメーター (XPS フロー)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_NUM_SLAVE_SLOTS ^(T)	1	整数 (1-16)	SI スロットの数
C_NUM_MASTER_SLOTS ^(T)	1	整数 (1-16)	MI スロットの数
C_FAMILY ^(T)	REQ	文字列	FPGA ファミリ
C_AXI_ID_WIDTH ^(T)	1	整数 (1-16)	AXI インターコネクト コアによって伝搬されるすべての ID 信号の幅
C_AXI_ADDR_WIDTH ^(C)	32	整数 (32)	全 SI スロットおよび全 MI スロットのすべての ADDR 信号の幅
C_S_AXI_IS_INTERCONNECT ^(T)	{N}0b0	{N} Bit1	CDAM ロジックがインプリメントされているかどうかの判別に使用します。 0 = エンドポイント マスター デバイスに接続 1 = 別の AXI インターコネクト コアに接続
C_INTERCONNECT_DATA_WIDTH ^(O)	接続された最も幅の広い MI スロットと同じ	整数 (32, 64, 128, 256, 512, 1024)	インターコネクト内部の書き込みおよび読み出しデータバスのデータ幅
C_INTERCONNECT_ACLK_RATIO ^(T)	1	整数 (1-2147483647)	全 SI スロットおよび MI スロットに関連する AXI インターコネクト コア内部のクロック周波数倍率(ツールによって Hz 単位でインターコネクト クロックの周波数に設定)
C_AXI_SUPPORTS_USER_SIGNALS ^(O)	0	整数	USER 信号 (5 チャンネルすべて) を AXI インターコネクト コアに伝搬するかどうかを示します。 0 = 伝搬しない 1 = 伝搬する
C_AXI_AWUSER_WIDTH ^(O)	1	整数 (1-256)	AXI4 の全 SI スロットおよび全 MI スロットの AWUSER 信号の幅
C_AXI_ARUSER_WIDTH ^(O)	1	整数 (1-256)	AXI4 の全 SI スロットおよび全 MI スロットの ARUSER 信号の幅
C_AXI_WUSER_WIDTH ^(O)	1	整数 (1-256)	AXI4 の全 SI スロットおよび全 MI スロットの WUSER 信号の幅
C_AXI_RUSER_WIDTH ^(O)	1	整数 (1-256)	AXI4 の全 SI スロットおよび全 MI スロットの RUSER 信号の幅
C_AXI_BUSER_WIDTH ^(O)	1	整数 (1-256)	AXI4 の全 SI スロットおよび全 MI スロットの BUSER 信号の幅
C_AXI_CONNECTIVITY ^(T)	すべて 1	{M} Bit32 {N} Bit1	各 SI スロット (N) から各 MI スロット (M) まではスパス クロスバーで接続するかどうかを示します(クロスバー モードのインターコネクトのみに適用)。 0 = パス不要 1 = パス要

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
- U = ユーザー指定
- T = ツールで生成 (EDK が情報を生成し、値を設定)
- C = 定数
- O = ユーザーのオーバーライドによりツールで生成または TCL で自動生成
- N = コアの HDL では使用しない

表 6: グローバル コア パラメーター (XPS フロー) (続き)

パラメーター名	デフォルト 値	表示形式/範囲	説明
C_INTERCONNECT_CONNECTIVITY_MODE ^(U)	1	整数 (0, 1)	インターコネクト アーキテクチャを定義します。 0 = 共有アクセス (エリア最適化) 1 = クロスバー (パフォーマンス最適化)
C_RANGE_CHECK ^(O)	次の場合は ON (1)。 C_NUM_MASTER_SLOTS>1 である場合、または、 C_M_AXI_BASE/HIGH_ADDR が 1 よりも大きい範囲を 定義する場合、または MI スロットのいずれかが AXI4-Lite である場合 かつ SI スロットのいずれかが AXI4-Lite でない場合、 または、 MI スロットのいずれかで C_M_AXI_SECURE が 設定されている場合。 上記以外は OFF (0)。	整数 (0, 1)	インターコネクト コアが各種トランザクション エラー状態を検出するかどうかを指定します。 0 (OFF) = DECERR の状態を検出しない。詳細は 52 ページの「デコード エラー検出」を参照。 1 (ON) = トランザクション エラーをトラップし、DECERR 応答を生成

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
 U = ユーザー指定
 T = ツールで生成 (EDK が情報を生成し、値を設定)
 C = 定数
 O = ユーザーのオーバーライドによりツールで生成または TCL で自動生成
 N = コアの HDL では使用しない

スレーブ インターフェイス パラメーター (XPS フロー)

表 7: スレーブ インターフェイス関連のパラメーター (XPS フロー)

パラメーター名	デフォルト 値	表示形式/範囲	説明
C_S_AXI_PROTOCOL ^(M)	{N} 0x00000000	{N} Bit32	接続されたマスター デバイスの AXI プロトコル 0 = SI スロットは AXI4 1 = SI スロットは AXI3 2 = SI スロットは AXI4-Lite
C_S_AXI_DATA_WIDTH ^(M)	{N} 0x00000020	{N} Bit32 (0x00000020、 0x00000040、 0x00000080、 0x00000100、 0x00000200、 0x00000400)	各 SI スロットの S_AXI_WDATA および S_AXI_RDATA の有効幅 (AXI4-Lite の SI ス ロットでは 0x20 の必要がある)
C_S_AXI_BASE_ID ^(I,O)	{N} 0x00000000	{N} Bit32 (0-0xFFFF)	各 SI スロットのベース ID (N-1:0)
C_S_AXI_THREAD_ID_WIDTH ^(M)	{N} 0x00000000	{N} Bit32 (0-0x10)	各 SI スロットの ID の下位にある可変ビットの 数 (N-1:0)。それぞれの値は <= C_AXI_ID_WIDTH の必要があります。
C_S_AXI_SINGLE_THREAD ^(I, U)	{N} 0b0	{N} Bit1	SI スロットによる ID スレッドのサポート 0 = Outstanding スレッド ID 値を複数個受け入 れる (パフォーマンス最適化) 1 = Outstanding スレッド ID 値を一度に 1 つの み受け入れる (エリア最適化)

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
 M = 値は接続されたマスター デバイス上に存在するパラメーターからコピー
 T = ツールで生成 (EDK が情報を生成し、値を設定)
 C = 定数
 O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)
 N = コアの HDL では使用しない
 U = ユーザー指定

表 7: スレーブ インターフェイス関連のパラメーター (XPS フロー) (続き)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_S_AXI_ACLK_RATIO ^(I,T)	{N}0x00000001	{N} Bit32 (0x1-0x7FFFFFFF)	内部インターコネクトに関連する各 SI スロットのクロック周波数倍率 (同期時) (ツールによって Hz 単位で SI クロックの周波数に設定)
C_S_AXI_IS_ACLK_ASYNC ^(I,O)	{N}d、 各 SI スロットのデフォルト値 d は (C_S_AXI_ACLK_RATIO[slot] : C_INTERCONNECT_ACLK_RATIO) の比率が 1:k または k:1 の場合は 0、それ以外の場合は 1。 k は 1 ~ 16 の整数。	{N} Bit1	SI スロットのクロックが、AXI インターコネクトのネイティブ クロックに同期か非同期かを示します。 0 = ネイティブ クロックに同期 1 = ネイティブ クロックに非同期
C_S_AXI_ARB_PRIORITY ^(I,U)	{N}0x00000000	{N} Bit32 (0x00000000 - 0x0000000f)	各 SI スロット間のアービトレーション優先順位。値が大きいくほど、優先順位が高いことを示します。値が 0 のすべてのスロットはラウンドロビンアービトレーションの対象になります。
C_S_AXI_WRITE_ACCEPTANCE ^(I,U)	{M}0x00000001	{M} Bit32 (0x1 - 0x20)	AXI の SI スロットが生成できるデータ アクティブな書き込みトランザクションの数
C_S_AXI_READ_ACCEPTANCE ^(I,U)	{M}0x00000001	{M} Bit32 (0x1 - 0x20)	AXI の SI スロットが生成できるアクティブな読み出しトランザクションの数
C_S_AXI_SUPPORTS_WRITE ^(M)	{N}0b1	{N} Bit1	各 SI スロットが書き込み関連チャネルを使用するかどうかを示します。 0 = 読み出し専用 1 = AW チャネル、W チャネル、および B チャネルを使用
C_S_AXI_SUPPORTS_READ ^(M)	{N}0b1	{N} Bit1	各 SI スロットが読み出し関連チャネルを使用するかどうかを示します。 0 = 書き込み専用 1 = AR チャネルと R チャネルを使用
C_S_AXI_SUPPORTS_NARROW_BURST ^(M,N)	{N}0b1	{N} Bit1	接続されたマスター デバイスが幅の狭いバーストを生成できるかどうかを示します。 0 = すべてのバーストはデータ幅と同じサイズであり、常に A*CACHE[1]=1 (単一ビートの転送には当てはまらない) 1 = 幅の狭いバーストを生成できる、または A*CACHE[1] をディアサート可
C_S_AXI_WRITE_FIFO_DEPTH ^(I,U)	{N}0x00000000	{N} Bit32 (0x00000000、0x00000020、0x00000200)	各 SI スロットの SI 側書き込みデータ FIFO (W チャネル アービトレーションの前) の深さ
C_S_AXI_WRITE_FIFO_DELAY ^(I,U)	{N}0b0	{N} Bit1	パケット FIFO 書き込み動作。バースト全体が SI 側の各 SI スロットの書き込みデータ FIFO に格納されるまでアービタへの AWVALID 発行を遅延させます (対応するスロットの C_S_AXI_WRITE_FIFO_DEPTH = 0x200 とする必要がある)。

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
M = 値は接続されたマスター デバイス上に存在するパラメーターからコピー
T = ツールで生成 (EDK が情報を生成し、値を設定)
C = 定数
O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)
N = コアの HDL では使用しない
U = ユーザー指定

表 7: スレーブ インターフェイス関連のパラメーター (XPS フロー) (続き)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_S_AXI_READ_FIFO_DEPTH (I,U)	{N}0x00000000	{N} Bit32 (0x00000000、 0x00000020、 0x00000200)	各 SI スロットの SI 側読み出しデータ FIFO (R チャンネルルーティングの後) の深さ
C_S_AXI_READ_FIFO_DELAY (I,U)	{N}0b0	{N} Bit1	パケット FIFO 読み出し動作。SI 側の各 SI ス ロットの読み出しデータ FIFO に全バースト長 を格納するのに十分な空きが生じるまで、アー ビタへの ARVALID 発行を遅延させます (対応 するスロットの C_S_AXI_READ_FIFO_DEPTH = 0x200 とす る必要がある)。
C_S_AXI_AW_REGISTER (I,U)	{N}0x00000000	{N} Bit32	各 SI スロット インターフェイスの AW チャン ネルにレジスタ スライスを挿入します。 0 = バイパス 1 = 完全レジスタ化 7 = 軽量化 8 = 自動
C_S_AXI_AR_REGISTER (I,U)	{N}0x00000000	{N} Bit32	各 SI スロット インターフェイスの AR チャン ネルにレジスタ スライスを挿入します。 0 = バイパス 1 = 完全レジスタ化 7 = 軽量化 8 = 自動
C_S_AXI_W_REGISTER (I,U)	{N}0x00000000	{N} Bit32	各 SI スロット インターフェイスの W チャン ネルにレジスタ スライスを挿入します。 0 = バイパス 1 = 完全レジスタ化 7 = 軽量化 8 = 自動
C_S_AXI_R_REGISTER (I,U)	{N}0x00000000	{N} Bit32	各 SI スロット インターフェイスの R チャン ネルにレジスタ スライスを挿入します。 0 = バイパス 1 = 完全レジスタ化 7 = 軽量化 8 = 自動
C_S_AXI_B_REGISTER (I,U)	{N}0x00000000	{N} Bit32	各 SI スロット インターフェイスの B チャン ネルにレジスタ スライスを挿入します。 0 = バイパス 1 = 完全レジスタ化 7 = 軽量化 8 = 自動

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
- M = 値は接続されたマスター デバイス上に存在するパラメーターからコピー
- T = ツールで生成 (EDK が情報を生成し、値を設定)
- C = 定数
- O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)
- N = コアの HDL では使用しない
- U = ユーザー指定

マスター インターフェイス パラメーター (XPS フロー)

表 8: マスター インターフェイス関連のパラメーター (XPS フロー)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_M_AXI_PROTOCOL ^(S)	{M} 0x00000000	{M} Bit32	接続されたスレーブ デバイスの AXI プロトコル 0 = MI スロットは AXI4 1 = MI スロットは AXI3 2 = MI スロットは AXI4-Lite
C_M_AXI_DATA_WIDTH ^(S)	{M} 0x00000020	{M} Bit32 (0x00000020、 0x00000040、 0x00000080、 0x00000100、 0x00000200、 0x00000400)	各 MI スロットの M_AXI_WDATA および M_AXI_RDATA の有効幅 (AXI4-Lite の MI スロッ トでは 0x20 の必要がある)
C_M_AXI_BASE_ADDR ^(I,U)	{M} ({16} 0xffffffff_ffffffff)	{M} ({16} Bit64)	各 MI スロット (M-1:0) の各範囲のベース アドレス (15:0)。未使用範囲にはベース アドレスを 0xffffffff_ffffffff に設定します。
C_M_AXI_HIGH_ADDR ^(I,U)	{M} ({16} 0x 00000000_ 00000000)	{M} ({16} Bit64)	各 MI スロット (M-1:0) の各範囲の高次アドレス (15:0)。未使用範囲では、高次アドレスを 0x00000000_00000000 に設定します。
C_M_AXI_ACLK_RATIO ^(I,T)	{M} 0x 00000001	{M} Bit32 (0x1- 0x7FFFFFFF)	内部 AXI インターコネクト コアに関連する各 MI スロットのクロック周波数倍率 (同期時) (ツールによって MI 単位で MI クロックの周波数 に設定)
C_M_AXI_IS_ACLK_ASYNC ^(I,O)	{M} d, 各 MI スロットの デフォルト値 d は (C_M_AXI_ACLK_ _RATIO[slot] : C_INTERCONNECT_ _ACLK_RATIO) の比率が 1:k または k:1 の場合は 0、 それ以外の場合は 1。 k は 1 ~ 16 の整数。	{M} Bit1	MI スロットのクロックが、AXI インターコネク トのネイティブ クロックに同期か非同期かを示 します。 0 = MI スロットのクロックは同期 1 = MI スロットのクロックは非同期
C_M_AXI_SUPPORTS_WRITE ^(S)	{M} 0b1	{M} Bit1	各 MI スロットが書き込み関連チャネルを使用す るかどうかを示します。 0 = 読み出し専用 1 = AW チャネル、W チャネル、および B チャネ ルを使用
C_M_AXI_SUPPORTS_READ ^(S)	{M} 0b1	{M} Bit1	各 MI スロットが読み出し関連チャネルを使用す るかどうかを示します。 0 = 書き込み専用 1 = AR チャネルと R チャネルを使用
C_M_AXI_WRITE_ISSUING ^(I,U)	{M} 0x00000001	{M} Bit32 (0x1 - 0x20)	AXI4 の MI スロットが生成できるデータ アク ティブな書き込みトランザクションの数
C_M_AXI_READ_ISSUING ^(I,U)	{M} 0x00000001	{M} Bit32 (0x1 - 0x20)	AXI4 の MI スロットが生成できるアクティブな 読み出しトランザクションの数

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
- S = 値は接続されたスレーブ デバイス上に存在するパラメーターからコピー
- U = ユーザー指定
- T = ツールで生成 (EDK が情報を生成し、値を設定)
- C = 定数
- N = コアの HDL では使用しない
- O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)

表 8: マスター インターフェイス関連のパラメーター (XPS フロー) (続き)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_M_AXI_SECURE (I,U)	{M}0b0	{M} Bit1	各 MI スロットがセキュアなスレーブ デバイス (TrustZone のセキュア アクセスを許可) に接続されているかどうかを示します。 0 = セキュアでないスレーブ デバイス 1 = セキュアなスレーブ デバイス
C_M_AXI_SUPPORTS_NARROW_BURST (S,N)	{M}0b1	{M} Bit1	接続されているスレーブ デバイスが、転送サイズがデータ幅よりも小さいバーストをサポートするよう構成されているかどうかを示します。 0 = 接続されているスレーブ デバイスが、MI スロットのデータ幅と SIZE とが異なるバーストを許容しない (単一バーストの転送には該当しない) 1 = 接続されているスレーブ デバイスが幅の狭いバーストをサポートする
C_M_AXI_WRITE_FIFO_DEPTH (I,U)	{M}0x00000000	{M} Bit32 (0x00000000、 0x00000020、 0x00000200)	各 MI スロットの MI 側書き込みデータ FIFO (W チャネル ルーティングの後) の深さ 0x0 = FIFO なし 0x20 = 深さ 32 の LUT RAM ベース FIFO 0x200 = 深さ 512 のブロック RAM ベース FIFO
C_M_AXI_WRITE_FIFO_DELAY (I,U)	{M}0b0	{M} Bit1	パケット FIFO 書き込み動作。バースト全体が MI 側の各 MI スロットの書き込みデータ FIFO に格納されるまでアービタへの M_AXI_AWVALID 発行を遅延させます (対応するスロットの C_M_AXI_WRITE_FIFO_DEPTH = 0x200 とする必要がある)。
C_M_AXI_READ_FIFO_DEPTH (I,U)	{M}0x 00000000	{M} Bit32 (0x00000000、 0x00000020、 0x00000200)	各 MI スロットの MI 側読み出しデータ FIFO (R チャネル アービトレーションの前) の深さ 0x0 = FIFO なし 0x20 = 深さ 32 の LUT RAM ベース FIFO 0x200 = 深さ 512 のブロック RAM ベース FIFO
C_M_AXI_READ_FIFO_DELAY (I,U)	{M}0b0	{M} Bit1	パケット FIFO 読み出し動作。MI 側の各 MI スロットの読み出しデータ FIFO に全バースト長を格納するのに十分な空きが生じるまで、アービタへの M_AXI_ARVALID 発行を遅延させます (対応するスロットの C_M_AXI_READ_FIFO_DEPTH = 0x200 とする必要がある)。
C_M_AXI_AW_REGISTER (I,U)	{M}0x00000000	{M} Bit32	各 MI スロット インターフェイスの AW チャネルにレジスタ スライスを挿入します。 0 = バイパス、1 = 完全レジスタ化 7 = 軽量化、8 = 自動
C_M_AXI_AR_REGISTER (I,U)	{M}0x00000000	{M} Bit32	各 MI スロット インターフェイスの AR チャネルにレジスタ スライスを挿入します。 0 = バイパス、1 = 完全レジスタ化 7 = 軽量化、8 = 自動
C_M_AXI_W_REGISTER (I,U)	{M}0x00000000	{M} Bit32	各 MI スロット インターフェイスの W チャネルにレジスタ スライスを挿入します。 0 = バイパス、1 = 完全レジスタ化 7 = 軽量化、8 = 自動

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
- S = 値は接続されたスレーブ デバイス上に存在するパラメーターからコピー
- U = ユーザー指定
- T = ツールで生成 (EDK が情報を生成し、値を設定)
- C = 定数
- N = コアの HDL では使用しない
- O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)

表 8: マスター インターフェイス関連のパラメーター (XPS フロー) (続き)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_M_AXI_R_REGISTER ^(I,U)	{M}0x00000000	{M} Bit32	各 MI スロット インターフェイスの R チャネルにレジスタ スライスを挿入します。 0 = バイパス、1 = 完全レジスタ化 7 = 軽量化、8 = 自動
C_M_AXI_B_REGISTER ^(I,U)	{M}0x00000000	{M} Bit32	各 MI スロット インターフェイスの B チャネルにレジスタ スライスを挿入します。 0 = バイパス、1 = 完全レジスタ化 7 = 軽量化、8 = 自動

注記:

- I = 接続されたすべてのマスター デバイスで有効な固有のパラメーター
S = 値は接続されたスレーブ デバイス上に存在するパラメーターからコピー
U = ユーザー指定
T = ツールで生成 (EDK が情報を生成し、値を設定)
C = 定数
N = コアの HDL では使用しない
O = ユーザーのオーバーライドにより、ツールで生成 (EDK が情報を生成して値を設定。ユーザーはオーバーライド可能)

グローバル パラメーター (CORE Generator フロー)

表 9: グローバル パラメーター (CORE Generator フロー)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_NUM_SLAVE_PORTS	2	整数 (1-16)	スレーブ インターフェイスの数
C_FAMILY	REQ	文字列	FPGA ファミリ
C_THREAD_ID_WIDTH	0	整数 (0-8)	すべてのスレーブ インターフェイス上のサンプリング対象の ID ビットの数 (存在する場合)
C_THREAD_ID_PORT_WIDTH	1	整数 (1-8)	すべてのスレーブ インターフェイス上の ID 信号の幅 (0 を取れない点を除き、C_THREAD_ID_WIDTH と同様)
C_AXI_ADDR_WIDTH	32	整数 (12-64)	SI および MI すべての全 ADDR 信号の幅
C_INTERCONNECT_DATA_WIDTH	C_M00_AXI_DATA_WIDTH と同じ	整数 (32、64、128、 256、512、1024)	インターコネクト内部の書き込みおよび読み出しデータバスのデータ幅

スレーブ インターフェイス パラメーター (CORE Generator フロー)

表 10 : スレーブ インターフェイス関連のパラメーター (CORE Generator フロー)

パラメーター名	デフォルト値	表示形式/範囲	説明
C_Snn_AXI_DATA_WIDTH	32	整数 (32、64、128、256、512、1024)	Snn_AXI_WDATA および Snn_AXI_RDATA 信号の幅
C_Snn_AXI_ACLK_RATIO	"1:1"	文字列 ("1:16"... "1:1"... "16:1")	内部インターコネクトに関連する各 SI のクロック周波数倍率 (同期時)
C_Snn_AXI_IS_ACLK_ASYNC	0	Bit1	SI のクロックが、AXI インターコネクトのネイティブ クロックに同期か非同期かを示します。 0 = ネイティブ クロックに同期 1 = ネイティブ クロックに非同期
C_Snn_AXI_ARB_PRIORITY	0	整数 (0 ~ 15)	各 SI 間のアービトレーション優先順位。値が大きいほど、優先順位が高いことを示します。値が 0 のすべての SI はラウンドロビン アービトレーションの対象になります。
C_Snn_AXI_WRITE_ACCEPTANCE	1	整数 (1 - 32)	AXI の SI スロットが生成できるデータ アクティブな書き込みトランザクションの数
C_Snn_AXI_READ_ACCEPTANCE	1	整数 (1 - 32)	AXI の SI スロットが生成できるアクティブな読み出しトランザクションの数
C_Snn_AXI_READ_WRITE_SUPPORT	"READ/WRITE"	文字列 ("READ/WRITE", "READ-ONLY", "WRITE-ONLY")	各 SI が書き込み関連チャネルまたは読み出し関連チャネルを使用するかどうかを示します。
C_Snn_AXI_WRITE_FIFO_DEPTH	0	整数 (0、32、512)	SI 側書き込みデータ FIFO (W チャネル多重化の前) の深さ 0 = FIFO なし 32 = 深さ 32 の LUT RAM ベース FIFO 512 = 深さ 512 のブロック RAM ベース FIFO
C_Snn_AXI_WRITE_FIFO_DELAY	0	Bit1	パケット FIFO 書き込み動作。バースト全体が SI 側の書き込みデータ FIFO に格納されるまでアービタへの AWVALID 発行を遅延させます (C_Snn_AXI_WRITE_FIFO_DEPTH = 512 とする必要がある)。
C_Snn_AXI_READ_FIFO_DEPTH	0	整数 (0、32、512)	SI 側読み出しデータ FIFO (R チャネルルーティングの後) の深さ 0 = FIFO なし 32 = 深さ 32 の LUT RAM ベース FIFO 512 = 深さ 512 のブロック RAM ベース FIFO
C_Snn_AXI_READ_FIFO_DELAY	0	Bit1	パケット FIFO 読み出し動作。SI 側の読み出しデータ FIFO に全バースト長を格納するのに十分な空きが生じるまで、アービタへの ARVALID 発行を遅延させます (C_Snn_AXI_READ_FIFO_DEPTH = 512 とする必要がある)。
C_Snn_AXI_REGISTER	0	Bit1	すべての SI チャネルにレジスタ スライスを挿入します。W および R チャネルでは完全レジスタ化、AW、AR、および B チャネルでは軽量化です。

マスター インターフェイス パラメーター (CORE Generator フロー)

表 11: マスター インターフェイス関連のパラメーター (CORE Generator フロー)

パラメーター名	デフォルト 値	表示形式/範囲	説明
C_M00_AXI_DATA_WIDTH	32	整数 (32、64、128、256、512、1024)	M00_AXI_WDATA および M00_AXI_RDATA 信号の幅
C_M00_AXI_ACLK_RATIO	"1:1"	文字列 ("1:16" ... "1:1" ... "16:1")	内部インターコネクトに関連する MI のクロック周波数倍率 (同期時)。
C_M00_AXI_IS_ACLK_ASYNC	0	Bit1	MI のクロックが、AXI インターコネクトのネイティブ クロックに同期しているか非同期的なかを示します。 0 = ネイティブ クロックに同期 1 = ネイティブ クロックに非同期的
C_Mnn_AXI_READ_WRITE_SUPPORT	"READ/WRITE"	文字列 ("READ/WRITE"、 "READ-ONLY"、 "WRITE-ONLY")	MI が書き込み関連チャネルまたは読み出し関連チャネルを使用するかどうかを示します。
C_M00_AXI_WRITE_ISSUING	1	整数 (1 - 32)	MI が発行できるデータ アクティブな書き込みトランザクションの数。
C_M00_AXI_READ_ISSUING	1	整数 (1 - 32)	MI が発行できるアクティブな読み出しトランザクションの数。
C_M00_AXI_WRITE_FIFO_DEPTH	0	整数 (0、32、512)	MI 側書き込みデータ FIFO (W チャネル多重化の後) の深さ 0 = FIFO なし 32 = 深さ 32 の LUT RAM ベース FIFO 512 = 深さ 512 のブロック RAM ベース FIFO
C_M00_AXI_WRITE_FIFO_DELAY	0	Bit1	パケット FIFO 書き込み動作。バースト全体が MI 側の書き込みデータ FIFO に格納されるまで M00_AXI_AWVALID の発行を遅延させます (C_M00_AXI_WRITE_FIFO_DEPTH = 512 とする必要がある)。
C_M00_AXI_READ_FIFO_DEPTH	0	整数 (0、32、512)	MI 側読み出しデータ FIFO (R チャネル ルーティングの前) の深さ 0 = FIFO なし 32 = 深さ 32 の LUT RAM ベース FIFO 512 = 深さ 512 のブロック RAM ベース FIFO
C_M00_AXI_READ_FIFO_DELAY	0	Bit1	パケット FIFO 読み出し動作。MI 側の読み出しデータ FIFO に全バースト長を格納するのに十分な空きが生じるまで、M00_AXI_ARVALID の発行を遅延させます (C_M00_AXI_READ_FIFO_DEPTH = 512 とする必要がある)。
C_M00_AXI_REGISTER	0	Bit1	すべての MI チャネルにレジスタ スライスを挿入します。W および R チャネルでは完全レジスタ化、AW、AR、および B チャネルでは軽量化です。

AXI インターコネクト パラメーターの使用方法

以降に示す表で、さまざまなデザイン パラメーターのより詳細な定義と使用方法 (値、要点、ほかのパラメーターとの相関) について説明します。パラメーターの説明では、AXI インターコネクト コアにあるパラメーターが、接続されたマスターおよびスレーブ デバイスのパラメーターからコピーまたは導出される状況についても解説します。

インターフェイス プロトコル

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_PROTOCOL	C_busif_PROTOCOL	
C_M_AXI_PROTOCOL		C_busif_PROTOCOL

*_PROTOCOL パラメーターは、AMBA AXI 仕様のインターフェイス サブプロトコル (AXI4、AXI3、または AXI4-Lite) を指定します。一般に、パラメーターは接続されたマスターおよびスレーブ IP の MPD に定数として指定されます。ただし、一部の IP ではプロトコルが設定可能であり (通常、AXI4 または AXI4-Lite)、ユーザーが選択できます。

AXI インターコネクト コアは Protocol パラメーターを次の目的で使用します。

- オプションのプロトコル変換モジュールを挿入する
- エラー検出ロジックを適用する
- AXI4-Lite の場合、使用しない AXI4 インターフェイス機能に関連するロジック リソースを節約する

ツールは、これらの値を接続されたマスターおよびスレーブから AXI インターコネクト コアにコピーします。

データ幅

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_DATA_WIDTH	C_busif_DATA_WIDTH	
C_M_AXI_DATA_WIDTH		C_busif_DATA_WIDTH
C_INTERCONNECT_DATA_WIDTH		

C_S_AXI_DATA_WIDTH パラメーターは接続されたマスター デバイスの、C_M_AXI_DATA_WIDTH パラメーターは接続されたスレーブ デバイスの WDATA 信号と RDATA 信号の幅を示します。ツールは、これらの値を接続されたマスターおよびスレーブから AXI インターコネクト コアにコピーします。

C_INTERCONNECT_DATA_WIDTH パラメーターは、内部クロスバーのネイティブ データ幅を指定します。デフォルトの場合、ツールは接続されている MI スロットで最も広い幅に一致するようにこの値を設定します。ただし、ユーザーはサポートされている範囲内であれば任意の値で (接続されているデバイスの幅に関係なく) これより優先させることができます。

動作は次のとおりです。

- SI スロットの C_S_AXI_DATA_WIDTH の値が C_INTERCONNECT_DATA_WIDTH よりも小さい場合、AXI インターコネクト コアの SI 域の SI スロットからのパス (SI とクロスバーの間) にアップサイザー モジュールが挿入されます。
- C_S_AXI_DATA_WIDTH の値が C_INTERCONNECT_DATA_WIDTH よりも大きい場合、ダウンサイザー モジュールが SI 域に挿入されます。
- C_M_AXI_DATA_WIDTH の値が C_INTERCONNECT_DATA_WIDTH と異なる場合、適切な幅コンバーターが MI 域 (クロスバーと MI の間) に挿入されます。

データ幅コンバーターにより、異なる幅のデータを接続するために必要なデータの圧縮とシリアル化が行われ、これにより SI、クロスバー、MI の間のパスを流れるデータ幅が影響を受けます。

C_INTERCONNECT_DATA_WIDTH に十分大きな値を選択することで、データの帯域幅損失を防止できます。

たとえば、メモリ コントローラーのようにスピードが重要なスレーブの場合、スレーブにアクセスするすべてのマスターのデータ幅がスレーブよりも狭くても C_INTERCONNECT_DATA_WIDTH には、このスレーブの幅に合わせた値を選択できます。このように設定することで、AXI インターコネクト コアはデータの圧縮 (書き込みトランザクションの場合) またはデータのシリアル化 (読み出しトランザクションの場合) を SI 域の複数の SI スロット パスで同時に実行できます。その間、マスターより幅の広いスレーブ デバイスおよびクロスバーは、データ スループット レートがマスター デバイス 1 つで保てる値を超えるように周期的に維持します。

一方、C_INTERCONNECT_DATA_WIDTH に低い値を選択すると、スピードをそれほど重視しないデザインにおいてロジック リソースの使用率を削減できます。リソースの使用率を最小限に抑えるには、幅コンバーター (アップサイザーおよびダウンサイザー) の総数が最小になるように C_INTERCONNECT_DATA_WIDTH を設定します。

インターコネクトは、SI とクロスバー間、クロスバーと MI 間のいずれにおいても、1024 ビットから 32 ビットへの直接のダウンサイジングはサポートしていません。いずれかの SI が 1024 ビット幅の場合、C_INTERCONNECT_DATA_WIDTH は 32 より大きい値に設定します。MI が 32 ビット幅の場合、C_INTERCONNECT_DATA_WIDTH は 1024 より小さい値に設定します。

クロック周波数

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_ACLK_RATIO	C_busif_ACLK_RATIO (C_busif_ACLK_FREQ_HZ も)	
C_M_AXI_ACLK_RATIO		C_busif_ACLK_RATIO (C_busif_ACLK_FREQ_HZ も)
C_S_AXI_IS_ACLK_ASYNC	C_busif_IS_ACLK_ASYNC	
C_M_AXI_IS_ACLK_ASYNC		C_busif_IS_ACLK_ASYNC
C_INTERCONNECT_ACLK_RATIO		

XPS ツールは、各バス インターフェイスに関連付けられたクロック ポート (ACLK) に対応する CLK_FREQ_HZ プロパティ (パラメーター C_busif_ACLK_FREQ_HZ で表される場合があります) を使用して、エンベデッド システム内のクロック信号をトラッキングします。

AXI インターコネクト コアには、クロスバーとその他の内部モジュールを同期させるグローバル クロック ポート、INTERCONNECT_ACLK もあります。ツールは、接続されたマスターおよびスレーブすべての CLK_FREQ_HZ 値を調査して INTERCONNECT_ACLK ポートの周波数と比較し、SI スロット、MI スロットのそれぞれとクロスバー間のクロック周波数の関係を判断します。

クロック周波数の関係は、ツールによって次のように判断されます。

- 1:16 ~ 16:1 の範囲の整数比 (高速または低速) であれば、ツールは対応する IS_ACLK_ASYNC パラメーターをゼロ (同期) に設定します。それ以外の場合は、スロットは非同期としてタグ付けされます。
- ツールは C_INTERCONNECT_ACLK_RATIO パラメーターに、INTERCONNECT_ACLK ポートの CLK_FREQ_HZ プロパティの値を割り当てます。また、C_S_AXI_ACLK_RATIO と C_M_AXI_ACLK_RATIO には、それぞれ SI ACLK と MI ACLK ポートの CLK_FREQ_HZ の値を割り当てます。コア内では、対応する C_S/M_AXI_IS_ACLK_ASYNC = 0 である場合にのみ、これらのパラメーター間の比率のみ意味を持ちます。
- SI スロットが非同期 (C_S_AXI_IS_ACLK_ASYNC = 1) の場合、あるいはクロック倍率が INTERCONNECT_ACLK のものと異なる場合 (C_S_AXI_ACLK_RATIO != C_INTERCONNECT_ACLK_RATIO の場合)、AXI インターコネクト コアの SI 域 (SI スロットとクロスバーの間のパス上) にクロック変換モジュールが挿入されます。
- MI スロットが非同期であるか、クロック倍率が AXI インターコネクト コアと異なる場合は、クロック変換モジュールが MI 域 (クロスバーと MI の間) に挿入されます。
- C_S/M_AXI_IS_ACLK_ASYNC = 0 の場合、倍率の違いを解消するために同期クロック変換モジュールが使用されます。これによってレイテンシとリソースが最小化されると共に、クロック ドメイン境界をまたぐパスのタイミング制約を適切に保つことができます。
- C_S/M_AXI_IS_ACLK_ASYNC = 1 の場合、非同期 FIFO に基づくクロック変換モジュールが使用され、各クロック ドメインの信号間のタイミング関係がなくなります。

クロック変換モジュールにより、常にレイテンシが増加します。SI と MI の間のパスを通る際に SI と MI の両域でクロック変換モジュールを通過するのはリソースの無駄遣いです。可能な限りクロック周波数を選択して、SI 域と MI 域の両方でクロック変換モジュールを通過しないようにしてください。

システム内のクロック変換モジュール数を減らすために、AXI インターコネクト インスタンスをカスケード接続して、同程度のクロック速度のデバイスをグループ化すると効果があります。たとえば、周波数の低い AXI4-Lite スレーブのグループを、同じ低周波数のクロックが供給されている別の AXI インターコネクト コアに接続することで、クロック ドメインがまたがる箇所をカスケード接続されている AXI インターコネクト インスタンス間のパスにある 1 つのコンバーターに統合できます。

メモリ コントローラーなど、スピードが重要なデバイスを AXI インターコネクト コアに接続する場合、スピードが要求されるスレーブと同じクロック ソースで INTERCONNECT_ACLK ポートにクロック供給を行うことによって、ほとんどの場合最良のデータ スループットが得られます。

アドレス範囲

インターコネクト	接続マスター	接続スレーブ
C_M_AXI_BASE_ADDR		C_busif_BASEADDR または C_busif_RNGnn_BASEADDR
C_M_AXI_HIGH_ADDR		C_busif_HIGHADDR または C_busif_RNGnn_HIGHADDR
C_AXI_ADDR_WIDTH	C_busif_ADDR_WIDTH	C_busif_ADDR_WIDTH

未使用のアドレス範囲のすべてで、C_M_AXI_BASE_ADDR の対応する値 (C_AXI_ADDR_WIDTH で指定されている数またはそれ以上の下位ビット) をすべて 1、C_M_AXI_HIGH_ADDR の対応する値をすべて 0 に設定します。

使用するアドレス範囲については次のように設定します。

- 範囲のサイズ (HIGH_ADDR - BASE_ADDR + 1) は 4k 以上である
- 範囲のサイズは 2 のべき乗である
- BASE_ADDR は範囲のサイズの倍数 (アラインされた状態) である
- アドレス デコード テーブル全体 (全 MI スロット) でアドレス範囲の重複は不可

アドレス範囲に関するすべての制約は、AXI インターコネクト コアのコンフィギュレーションに使用されるツールで強制的に適用され、コア自体のエラー チェックでは強制適用されません。

2 つのインターコネクト インスタンスがカスケード接続されている場合、アップストリームのインターコネクトのアドレス デコーダーは、カスケード接続された MI スロットの複数のアドレス範囲を使用し、ダウンストリームのインターコネクト インスタンスからアクセス可能なすべてのダウンストリーム スレーブ デバイスのアドレス範囲の集合を表します。

異なるエンドポイントのスレーブ デバイスを表すには別々のアドレス領域を使用することが重要です。また、システム アドレス マップで隣接または近くの領域にマッピングされていたとしても、複数のスレーブ デバイスを 1 つのアドレス デコーダー領域に結合しないことも需要です。これにより、CDAM (Cyclic Dependency Avoidance Method) を使用して、各 SI スロットで受信した ID スレッドのそれぞれは、一度に 1 つのエンドポイント スレーブ デバイスに対してのみ Outstanding 書き込み/読み出しトランザクションを生成できるというポリシーを適用できます。カスケード接続された MI スロットに複数のアドレス範囲を定義する方法は、23 ページの「複数アドレス範囲のサポート」と 23 ページの図 10 の例を参照してください。

エンベデッド ハードウェア システムでは 32 ビット アドレスのみをサポートしているため、C_AXI_ADDR_WIDTH の値は 32 (定数) です。

注記：AXI インターコネクト コアはアドレス マップの再割り当てをサポートしていません。したがって、マルチポートのスレーブ デバイス (マルチポート メモリ コントローラーなど) をシステムで使用する (複数のバス インターフェイスが同一の、または重複したアドレス範囲を共有する) 場合、これらの複数バス インターフェイスは AXI インターコネクトの別々のインスタンスに接続する必要があります。また、これらの AXI インターコネクト インスタンスを相互にカスケード接続しないでください。ツールでは、同じインターコネクト内で複数のアドレス範囲は重複できないという規則が強制的に適用されます。

ID 範囲

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_BASE_ID	C_INTERCONNECT_busif_BASE_ID	
C_S_AXI_THREAD_ID_WIDTH	C_busif_SUPPORTS_THREADS, C_busif_THREAD_ID_WIDTH	
C_AXI_ID_WIDTH		C_busif_ID_WIDTH

ID 値により、正しい SI スロットと、接続されたマスター デバイスに応答が戻されます。

深さが 1 よりも大きいトランザクション順序変更をサポートするマスター デバイスでは、AWID および ARID 出力を異なる値で駆動することにより、トランザクションの複数の「スレッド」を発行できます。

さまざまな ID パラメーターは、次のように関連します。

- C_S_AXI_THREAD_ID_WIDTH ベクター パラメーターは、各マスター デバイスが生成する ID のビット数を指定します。順序変更の深さが 1 のマスターは ID 信号を生成せず、対応する C_S_AXI_THREAD_ID_WIDTH の値を 0 に設定します。SI スロットでサンプリングされるすべてのスレッド ID ビットは、インターコネクト コア全体で使用され、MI に伝搬される完全なトランザクション ID 信号の下位ビットとして使用されます。
- C_AXI_ID_WIDTH グローバル パラメーターはインターコネクト全体で使用され、すべての MI スロットによって伝搬される完全なトランザクション ID 信号の幅を指定します。これはすべての SI スロットを一意に区別するのに十分な上位ビット (マスター ID) を含んでいる必要があります。
- C_S_AXI_BASE_ID パラメーターは、各 SI スロットに対応するベース (最小) ID の値を定義します。
- C_S_AXI_BASE_ID パラメーター値のうち、C_S_AXI_THREAD_ID_WIDTH で示される下位ビット位置の値はすべて 0 でなければなりません。
- C_S_AXI_BASE_ID の残りの上位ビットは「マスター ID」とみなされ、各 SI スロットから伝搬されるすべてのトランザクションで一定です。
- インターコネクト全体で使用され、すべての MI スロットによって伝搬される完全なトランザクション ID 信号は、SI スロットによってサンプリングされたスレッド ID ビット (存在する場合) と、SI スロットの C_S_AXI_BASE_ID の値の論理和によって生成されます。

EDK ツールは AXI インターコネクトの C_AXI_ID_WIDTH パラメーターに必要とされる値を判断し、各 SI スロットで検出した C_S_AXI_BASE_ID および C_S_AXI_THREAD_ID_WIDTH パラメーターの値に基づいてすべての SI スロットに一意のマスター ID 値を割り当てます。

AXI インターコネクト コアは C_S_AXI_BASE_ID パラメーターと C_S_AXI_THREAD_ID_WIDTH パラメーターを使用して、R チャネルおよび B チャネル上の応答を適切な SI スロットに戻すデコード ロジックをインプリメントします。

SI スロットが別の (アップストリームの) AXI インターコネクト インスタンスの MI スロットに接続されている場合、そのスロットの ID 信号で受信できる ID 値の範囲は、アップストリーム AXI インターコネクト コアの全 SI スロット (接続された MI スロットへの接続パスを持つもの) にあるすべての ID 値を含みます。アップストリーム AXI インターコネクト コアで生成された ID 信号は、接続されたマスター デバイスのスレッド ID ビットとして扱われます。

ほかのマスター デバイスと同様に、ダウンストリームの AXI インターコネクトはカスケード接続された SI スロットからサンプリングされた ID 信号の前に一意のマスター ID を付加します。これにより、カスケード接続された AXI インターコネクト トポロジを伝搬するにつれて ID 幅が大きくなります。マスター ID が一致する応答すべてがアップストリームの AXI インターコネクトに戻されます。

アドレス範囲と同様に、BASE_ID パラメーターの下位ビット (THREAD_ID_WIDTH によって指定) がすべて 0 でなければならぬことから、各 ID 範囲はアラインされた 2 のべき乗で表されるサイズになります。

ID デコード テーブル全体 (全 SI スロット) で、ID 範囲は重複できません。アドレス範囲とは異なり、ID 範囲には最小サイズとして 1 が規定されています。

AXI インターコネクトに対するエンドポイント マスター デバイスのスレッド ID 幅パラメーターは次のように指定します。

- AXI インターコネクト コアの SI スロットの C_S_AXI_THREAD_ID_WIDTH 値が 0 の場合、対応する ID 範囲のサイズは単一の ID 値となり、SI スロットでサンプリングされる ID 信号はありません。
- 接続されたマスター デバイス上では、対応する C_busif_THREAD_ID_WIDTH パラメーターを使用して IP インターフェイス上の ID 信号の物理的なビット幅を決定することが多いため、この値を 1 より小さい値に設定しないでください。
- C_busif_SUPPORTS_THREADS パラメーターはエンドポイント マスター デバイスで使用され、マスターのスレッドの有効幅が 0 であることを示します。
- 接続されたマスターで C_busif_SUPPORTS_THREADS = 0 のとき、AXI インターコネクト コア上の対応する SI スロットの C_S_AXI_THREAD_ID_WIDTH を 0 に設定します。

トランザクションの受け入れ限度と発行限度

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_WRITE_ACCEPTANCE	C_INTERCONNECT_busif_WRITE_ISSUING	
C_S_AXI_READ_ACCEPTANCE	C_INTERCONNECT_busif_READ_ISSUING	
C_M_AXI_WRITE_ISSUING		C_INTERCONNECT_busif_WRITE_ACCEPTANCE
C_M_AXI_READ_ISSUING		C_INTERCONNECT_busif_READ_ACCEPTANCE

C_S_AXI_WRITE_ACCEPTANCE パラメーターと C_S_AXI_READ_ACCEPTANCE パラメーターは、その時点でクロスバーが受け入れるタイプ別の **Outstanding** トランザクション数の上限を **SI** スロットごとに設定します。クロスバーは、**SI** スロットの **ACCEPTANCE** 上限と、異なる **AWID/ARID** 値の数 ($2 \times \text{THREAD_ID_WIDTH}$) のうち小さい方に基づいてトランザクション カウンターを維持し、同時に処理するスレッドの最大数に対応します。**ACCEPTANCE** 上限パラメーターでは、クロスバーに到着する前の **SI** 域でアドレス チャンネルにインプリメントされる、レジスタ スライスやクロック変換モジュールなどのバッファ モジュールで受け入れて保管するアドレス転送の数を対象としていません。

C_M_AXI_WRITE_ISSUING パラメーターと C_M_AXI_READ_ISSUING パラメーターは、その時点でクロスバーが (**ID** 値を指定して) 発行するタイプ別の **Outstanding** トランザクション総数の上限を設定します。また、**ISSUING** 上限パラメーターでは、クロスバーが発行した後に **MI** 域でアドレス チャンネルにインプリメントされる、レジスタ スライスやクロック変換モジュールなどのバッファ モジュールで受け入れて保管するアドレス転送の数を対象としていません。

受け入れカウンターと発行カウンターのカウント条件は次のとおりです。

- 書き込みトランザクションは、クロスバーで **BVALID/BREADY** のハンドシェイクが完了したときに完了したとみなされます (カウンター値は 1 減少)。
- 読み出しトランザクションは、クロスバーで **Rlast** をアサートする **RVALID/RREADY** ハンドシェイクが完了したときに完了したとみなされます。

書き込み/読み出しトランザクションが、受け入れ限度に達した **SI** スロットまたは発行限度に達した **MI** スロットに対して送信されている場合にはアービトレーションの対象から外され、書き込み/読み出しアービタが停止せずにその他の適格な **SI** スロットへのアービトレーションを継続できるようにします。

ACCEPTANCE または **ISSUING** パラメーターの値を大きくすると、接続されたスレーブ デバイスで書き込み/読み出しコマンドのパイプライン処理が可能となるため、データ スループットが向上し、書き込み/読み出しデータ チャンネルでのアイドル サイクルを回避できます。ただし、パラメーター値を大きくしすぎると、複数のマスター デバイスが共有スレーブにアクセスする際に **HOL (Head-of-Line)** ブロッキングが発生します。

ツールで、接続されたマスターおよびスレーブ デバイスから **AXI** インターコネクト コアへパラメーター値をコピーする際には、次のことが起こります。

- 接続されたマスター デバイスの **ISSUING** パラメーターは、**AXI** インターコネクト コアの **SI** 上の **ACCEPTANCE** パラメーターにマッピングされます。
- 接続されたスレーブ デバイスの **ACCEPTANCE** パラメーターは、**AXI** インターコネクト コアの **MI** 上の **ISSUING** パラメーターにマッピングされます。

注記：AXI4-Lite の **SI** スロットと **MI** スロットでは、それぞれ **ACCEPTANCE** パラメーターと **ISSUING** パラメーターが無視され、一度に 1 つの **Outstanding** トランザクションのみが許可されます。

アービトレーションの優先順位

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_ARB_PRIORITY	C_INTERCONNECT_busif_ARB_PRIORITY	

原則として、対応する SI スロットの相対優先順位に基づくアービトレーションにより使用権が与えられます。

C_S_AXI_ARB_PRIORITY パラメーターには、0 から 15 までの優先順位の値を静的に設定できます。値が大きいくほど、優先順位が高くなります。優先順位が同じ場合は、次のように処理されます。

- 要求送出中の適格な SI スロットすべての優先順位がレベル 0 の場合、アービトレーションはラウンドロビンになります。
- 要求送出中の SI スロットのうち、最も高い優先順位の値が 0 よりも大きい場合、優先順位が同じスロットの間では、スロット番号が小さいものが優先されます。

書き込み/読み出しトランザクションが、受け入れ限度に達した SI スロットまたは発行限度に達した MI スロットに対して送信されている場合には一時的にアービトレーションの対象から外され、書き込み/読み出しアービタが停止せず、その他の適格な SI スロットへのアービトレーションを継続できるようにします (46 ページの「トランザクションの受け入れ限度と発行限度」を参照)。

また、SI スロットから MI スロットへのアクセスでデッドロックが発生する可能性がある場合、その SI は一時的にアービトレーションの対象外となります (24 ページの「サイクル依存性の回避」を参照)。

クロスバーの接続

インターコネクト	接続マスター	接続スレーブ
C_AXI_CONNECTIVITY		C_INTERCONNECT_busif_MASTERS
C_INTERCONNECT_CONNECTIVITY_MODE		
C_S_AXI_SINGLE_THREAD	C_INTERCONNECT_busif_SINGLE_THREAD	
C_S_AXI_IS_INTERCONNECT		

AXI インターコネクト コアは、C_INTERCONNECT_CONNECTIVITY_MODE パラメーターによって選択される 2 つのコネクティビティ アーキテクチャをサポートしています。

共有アクセス モードに設定した場合 (C_INTERCONNECT_CONNECTIVITY_MODE = 0)、AXI インターコネクト コアは一度に 1 つだけ Outstanding トランザクションを提供します。共有アクセス モードを使用すると、同時データ転送のパフォーマンスが不要なアプリケーションで、インターコネクトのクロスバー モジュールをインプリメントするために必要なリソースが最小限に抑えられます。

クロスバー モードに設定した場合 (C_INTERCONNECT_CONNECTIVITY_MODE = 1)、AXI インターコネクト コアは「スパースクロスバー」コネクティビティをサポートします。これにより、SI スロットと MI スロットとのアクセスをユーザーが指定できます。未使用のパスを無効にすることで、データパス多重化ロジックとアドレス デコード ロジックを削減でき、使用する FPGA リソースを減らしてタイミング パスを高速化できます。

クロスバー モードに設定した場合、残りのコネクティビティ パラメーターは次のように設定されます。

- スパース コネクティビティの情報は、AXI インターコネクト コアの C_AXI_CONNECTIVITY パラメーターで表されます。
- C_AXI_CONNECTIVITY の値は、接続されたすべてのスレーブ デバイスの C_INTERCONNECT_busif_MASTERS パラメーターの累計から導き出されます。
- 各スレーブ デバイスの C_INTERCONNECT_busif_MASTERS パラメーターは、そのスレーブにアクセスする必要のあるすべてのマスター デバイスを、そのインスタンス名とマスター インスタンスのバス インターフェイス名に従って列挙します。

XPS GUI を使用してシステムのバス接続を確立する場合、MHS (Microprocessor Hardware System) デザイン ファイルに C_INTERCONNECT_busif_MASTERS パラメーターが自動的に挿入されます。

C_S_AXI_IS_INTERCONNECT パラメーターもクロスバー接続に影響を与えます。このパラメーターは、別の AXI インターコネクトに駆動されている SI スロットがあるかどうかを示します。

axi2axi_connector コアを使用してアップストリームの AXI インターコネクトの MI スロットをダウンストリームの AXI インターコネクトの SI スロットに渡すことによって、2 つの AXI インターコネクト コアをカスケード接続できます。

ツールが SI スロット上に `axi2axi_connector` を検出した場合、一定の条件で `C_S_AXI_IS_INTERCONNECT` パラメーターの値が自動的に設定されます。ユーザーの操作は必要ありません。このパラメーターは、アップストリームの AXI インターコネクトで実行されるトランザクション スレッド制御ロジックの重複を可能な限り回避するために、ダウンストリームの AXI インターコネクトで使用されます。

`C_S_AXI_SINGLE_THREAD` パラメーターによって、一度に許可する 1 つ以上の **Outstanding** トランザクションを 1 つのスレッド ID だけからにすることで、スレッド制御ロジックが SI スロットごとの簡潔なものになります。この方法により、特に SI スロットがマスター デバイスまたはアップストリームのインターコネクトに接続され、非常に広い ID 信号が生成される場合に、リソース節約の効果が得られることがあります。

コネクティビティ パラメーターの相互関係を 表 12 に示します。

表 12 : コネクティビティ パラメーター間の関係

C_INTERCONNECT_CONNECTIVITY_MODE	C_S_AXI_SINGLE_THREAD	C_S_AXI_IS_INTERCONNECT	説明
0	x	x	共有アクセス アーキテクチャ インターコネクトは一度に 1 つのトランザクションを受け入れます。
1	1	x	SI スロットが単一スレッドです。 SI スロットは複数のトランザクションを受け入れますが、一度に 1 つのスレッド ID 値に限られます。
1	0	0	SI スロットが複数スレッドです。 SI スロットが一度に複数の ID スレッドのトランザクションを受け入れますが、宛先 MI スロットは 1 スレッドあたり 1 つに限られます。
1	0	1	SI スロットが複数スレッドです。 SI スロットが一度に複数の ID スレッドのトランザクションを受け入れ、1 スレッドあたり、アップストリームのインターコネクトが宛先エンドポイント スレーブ デバイスを 1 つのみ受け入れると想定します。

注記 : XPS ツールは、マスターおよびスレーブのインスタンスとそのバス インターフェイスが MHS ファイルに現れる順番に従って、連続する SI スロット番号を接続されたマスターに、連続する MI スロット番号を接続されたスレーブに自動的に割り当てます。マスター/スレーブのインスタンスやそのバス インターフェイス接続が追加、削除、置換された場合に、スロット番号の割り当てをユーザーが明示的に制御したり、割り当てられたスロット番号を反復デザイン時に保持したりするメカニズムはありません。AXI インターコネクト コアは、SI または MI の「空き」のサポートはしません。

読み出し専用インターフェイスと書き込み専用インターフェイス

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_SUPPORTS_WRITE	C_busif_SUPPORTS_WRITE	
C_S_AXI_SUPPORTS_READ	C_busif_SUPPORTS_READ	
C_M_AXI_SUPPORTS_WRITE		C_busif_SUPPORTS_WRITE
C_M_AXI_SUPPORTS_READ		C_busif_SUPPORTS_READ

一般に、読み出し専用/書き込み専用のパラメーターは接続されたマスター/スレーブ IP の MPD に定数として指定されます。しかし、変更可能な読み出し専用/書き込み専用動作 (ユーザーが選択可能) をサポートする IP もあります。読み出しおよび書き込みのサポートを無効にする場合、次の規則が適用されます。

- C_AXI_CONNECTIVITY パラメーターが特定の SI スロットおよび MI スロット ペア間のパスを無効にするのと同様に、SUPPORTS_WRITE パラメーターを無効にすると、対応する SI または MI スロットの AW、W、および B チャンルの接続パスが無効になります。
- SUPPORTS_READ パラメーターが無効の場合、対応する SI または MI スロットの AR チャンネルと R チャンネルの接続パスが無効になります。
- 未使用の書き込み/読み出しチャンネルを無効にすることで、データパス多重化ロジックやアドレス デコード ロジックを削減でき、使用する FPGA リソースを減らすと共にタイミング パスを高速化できます。
- 各 SI スロットおよび各 MI スロットについて、SUPPORTS_WRITE または SUPPORTS_READ (もしくはその両方) が有効の必要があります。

読み出しおよび書き込みトランザクションをサポートするよう構成できるスレーブ デバイスの場合、これらのデバイスにアクセスする接続されたマスター デバイスで検出された値に基づいて、ツールが対応するパラメーター値を設定します。

レジスタ スライス

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_AW_REGISTER	C_INTERCONNECT_busif_AW_REGISTER	
C_S_AXI_AR_REGISTER	C_INTERCONNECT_busif_AR_REGISTER	
C_S_AXI_W_REGISTER	C_INTERCONNECT_busif_W_REGISTER	
C_S_AXI_R_REGISTER	C_INTERCONNECT_busif_R_REGISTER	
C_S_AXI_B_REGISTER	C_INTERCONNECT_busif_B_REGISTER	
C_M_AXI_AW_REGISTER		C_INTERCONNECT_busif_AW_REGISTER
C_M_AXI_AR_REGISTER		C_INTERCONNECT_busif_AR_REGISTER
C_M_AXI_W_REGISTER		C_INTERCONNECT_busif_W_REGISTER
C_M_AXI_R_REGISTER		C_INTERCONNECT_busif_R_REGISTER
C_M_AXI_B_REGISTER		C_INTERCONNECT_busif_B_REGISTER

REGISTER パラメーターを設定すると、SI 域または MI 域で最も外側のレジスタ スライスが有効になります。各インターフェイス スロットのチャンネルのそれぞれが独立して有効化されます。レジスタ スライスを設ける主な目的は、1 レイテンシ サイクルを犠牲にしてシステム タイミングを改善することです。

モード選択は次のとおりです。

- **FULLY_REGISTERED** : 深さ 2 の FIFO バッファとしてインプリメントされるこのモードは、バブル サイクルを引き起こすことなく、チャンネル ソースやチャンネル デスティネーションのスロットルと **back-to-back** 転送をサポートします。
- **LIGHT_WEIGHT** : 単純な 1 段パイプライン レジスタとしてインプリメントされるこのモードは、タイミング パスを分離しつつリソースを最小化しますが、毎回の転送後必ず 1 バブル サイクルが発生します。
- **AUTOMATIC** : AXI4-Lite スロットでは、全チャンネルに **LIGHT_WEIGHT** を使用します。その他のプロトコルの場合、W チャンネルと R チャンネルには **FULLY_REGISTERED** を、AW、AR、および B チャンネルには **LIGHT_WEIGHT** を使用します。

データパス FIFO

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_WRITE_FIFO_DEPTH	C_INTERCONNECT_busif_WRITE_FIFO_DEPTH	
C_S_AXI_READ_FIFO_DEPTH	C_INTERCONNECT_busif_READ_FIFO_DEPTH	
C_M_AXI_WRITE_FIFO_DEPTH		C_INTERCONNECT_busif_WRITE_FIFO_DEPTH

インターコネクト	接続マスター	接続スレーブ
C_M_AXI_READ_FIFO_DEPTH		C_INTERCONNECT_busif_READ_FIFO_DEPTH
C_S_AXI_WRITE_FIFO_DELAY	C_INTERCONNECT_busif_WRITE_FIFO_DELAY	
C_S_AXI_READ_FIFO_DELAY	C_INTERCONNECT_busif_READ_FIFO_DELAY	
C_M_AXI_WRITE_FIFO_DELAY		C_INTERCONNECT_busif_WRITE_FIFO_DELAY
C_M_AXI_READ_FIFO_DELAY		C_INTERCONNECT_busif_READ_FIFO_DELAY

*_FIFO_DEPTH パラメーターを設定すると、SI 域または MI 域のデータ チャネル上のクロスバーのすぐ隣に FIFO バッファが挿入されます。パラメーターの値により、FIFO の深さが設定されます (0 = FIFO なし)。各 FIFO に保存されるデータの幅は、常にクロスバーのネイティブ データ幅 (C_INTERCONNECT_DATA_WIDTH) に一致します。

デフォルト設定の場合、FIFO バッファは W および R (または、そのいずれかの) データ チャネルのみに挿入されます。*_FIFO_DELAY パラメーターが有効な場合、対応する AW または AR アドレス チャネルには、深さ 32 の FIFO バッファ (分散 RAM に基づく) も挿入されます (書き込み応答チャネルには FIFO 機能がありません)。

パケット FIFO 書き込み動作を示す *_WRITE_FIFO_DELAY パラメーターが有効な場合、W チャネル FIFO が WLAST を受信するまで、AW FIFO からの AW チャネル転送読み出しは実行されません。これによって、書き込みバースト全体が FIFO に格納され転送の準備が整うまで書き込みトランザクションの発行が遅延されるため、低速の書き込みデータ ソースによるストールは発生しません。

パケット FIFO 読み出し動作を示す *_READ_FIFO_DELAY パラメーターが有効な場合、R チャネル FIFO 内の空き領域が ARLEN の示す全バースト長以上になるまで AR FIFO からの AR チャネル転送読み出しは実行されません。(「空き領域」とは、発行済みの AR コマンドによって占有されていない R チャネル FIFO 内の空きスペースを意味します)。これによって、読み出しバースト全体が FIFO に確実に格納できるようになるまで読み出しトランザクションの発行が遅延されるため、低速の読み出しデータ宛先によるストールは発生しません。

TrustZone セキュリティ

インターコネクト	接続マスター	接続スレーブ
C_M_AXI_SECURE		C_INTERCONNECT_busif_SECURE

C_M_AXI_SECURE パラメーターを設定すると、指定した (スレーブに接続されている) 各 MI スロットに一括で TrustZone セキュリティが適用されます。セキュア スロットとして設定された場合、セキュアな AXI アクセスのみが許可されます (S_AXI_AWPROT[1] または M_AXI_AWPROT[1] が 0)。セキュアでないアクセスはブロックされ、AXI インターコネクト コアは SI 上で DECERR 応答を生成します。接続されたスレーブ デバイスが独自の TrustZone セキュリティ機能を持つ場合は、通常 AXI インターコネクトで C_M_AXI_SECURE パラメーターを設定する必要はありません。

幅の狭いバーストのサポート

インターコネクト	接続マスター	接続スレーブ
C_S_AXI_SUPPORTS_NARROW_BURST	C_busif_SUPPORTS_NARROW_BURST	
C_M_AXI_SUPPORTS_NARROW_BURST		C_busif_SUPPORTS_NARROW_BURST

SI スロットで C_S_AXI_SUPPORTS_NARROW_BURST パラメーターを 0 に設定した場合、接続されたマスター デバイスは「行儀のよい」デバイスであり、データ転送の SIZE がインターフェイスのデータ幅よりも小さい複数ビット トランザクション (「幅の狭いバースト」) を決して発行しないことを表します。たとえば、64 ビット幅の WDATA/RDATA 信号を使用する「行儀のよい」マスターは、AWLEN/ARLEN > 0 のトランザクションの発行時、その AWSIZE/ARSIZE が常に 0x011 (8 バイト) になります (シングル ビット トランザクションの SIZE は任意)。接続された「行儀のよくない」マスターはすべて、C_S_AXI_SUPPORTS_NARROW_BURST パラメーターをデフォルトの 1 に保つ必要があります。

また、`C_S_AXI_SUPPORTS_NARROW_BURST = 0` の場合、マスター デバイスはどのトランザクションの `AWCACHE[1]` または `ARCACHE[1]` (「変更許可」ビット) もディASSERTしてはなりません。

これにより、**SI** スロットによって駆動されるバス上のアップサイザーによりデータが完全に圧縮され、「行儀のよい」**SI** スロットで生成されたすべてのトランザクションで、**MI** での幅の狭いバーストが生成されなくなります。

`C_S_AXI_SUPPORTS_NARROW_BURST = 0` に設定した **SI** スロットのいずれかで幅の狭いバーストが受信された場合、**AXI** インターコネクト コアは **DECERR** 応答の検出、送信のいずれも実行しません。これは幅の狭いバーストを回避するためのマスター **IP** のデザイン要件です。

MI スロットの `C_M_AXI_SUPPORTS_NARROW_BURST` パラメーターが **0** に設定されている場合、接続されたスレーブ デバイスが幅の狭いバーストを受信することがなく、これに関するデータの圧縮ロジックが省略できることを表します。

幅の狭いバーストに対応したコンフィギュレーションをサポートするスレーブ デバイスでは、スレーブにアクセスできる全マスター デバイスが「行儀よく」動作する (マスターの `C_busif_SUPPORTS_NARROW_BURST` が **0** に設定されている) 場合、**XPS** ツールにより `C_busif_SUPPORTS_NARROW_BURST` パラメーターが自動的に **0** に設定されます。

AXI インターコネクト コア自体が `SUPPORTS_NARROW_BURST` パラメーターのいずれかを使用することはありません。**AXI** インターコネクトの当バージョンのすべてのアップサイザーでは常に (要件どおりに変更許可ビットが設定されている場合) 複数ビット バーストを圧縮して、インターコネクトにより幅の狭いバーストが作成されないようにしています。

ユーザー信号

インターコネクト	接続マスター	接続スレーブ
<code>C_AXI_SUPPORTS_USER_SIGNALS</code>	<code>C_busif_SUPPORTS_USER_SIGNALS</code>	<code>C_busif_SUPPORTS_USER_SIGNALS</code>
<code>C_AXI_AWUSER_WIDTH</code>	<code>C_busif_AWUSER_WIDTH</code>	<code>C_busif_AWUSER_WIDTH</code>
<code>C_AXI_ARUSER_WIDTH</code>	<code>C_busif_ARUSER_WIDTH</code>	<code>C_busif_ARUSER_WIDTH</code>
<code>C_AXI_WUSER_WIDTH</code>	<code>C_busif_WUSER_WIDTH</code>	<code>C_busif_WUSER_WIDTH</code>
<code>C_AXI_RUSER_WIDTH</code>	<code>C_busif_RUSER_WIDTH</code>	<code>C_busif_RUSER_WIDTH</code>
<code>C_AXI_BUSER_WIDTH</code>	<code>C_busif_BUSER_WIDTH</code>	<code>C_busif_BUSER_WIDTH</code>

接続されたマスター デバイスとスレーブ デバイスでこれらのパラメーターを指定すると、これらのデバイスのインターフェイスに **USER** 信号 (どのチャンネルでも可) が含まれるかどうかと、含まれる場合は各チャンネルの **USER** 信号の幅が示されます。

接続されたマスター デバイスおよびスレーブ デバイスのパラメーター値を調べることで、ツールは **AXI** インターコネクト コアが (任意のチャンネルの) **SI** と **MI** 間の **USER** 信号を伝搬する必要があるかどうかと、その場合に **USER** 信号を各チャンネルに伝搬するために必要な最大幅を決定します。その後、ツールは接続されたマスターおよびスレーブで検知した値に基づいて、**AXI** インターコネクト コアに **USER** パラメーターの値を自動的に設定します。

一般に、**IP** インターフェイスの **USER** 信号の物理ビット幅はさまざまな `USER_WIDTH` パラメーターを使用して決定されるため、**1** よりも小さな値に設定しないでください。`C_AXI_SUPPORTS_USER_SIGNALS = 1` の場合、**USER** 信号の少なくとも **1** ビットは **5** つの **AXI** チャンネルのそれぞれに伝搬されます。

WUSER 信号と **RUSER** 信号の伝搬は、インターフェイスのデータ幅が **AXI** インターコネクトのネイティブ データ幅 (`C_INTERCONNECT_DATA_WIDTH`) と一致する **SI** スロットと **MI** スロットに対してのみサポートされます。**RUSER** 信号と **WUSER** 信号は、アップサイザーおよびダウンサイザーにより常にブロック (すべて **0** に強制変換) されます。トランザクションが分割される可能性があるため、ダウンサイザーまたは **AXI3** プロトコル コンバーターは **BUSER** 信号を伝搬しません。**AWUSER** および **ARUSER** 信号の伝搬は、データ幅にかかわらず、すべての **SI** および **MI** スロットでサポートされています。

デコード エラー検出

AXI インターコネクト パラメーター	説明
C_RANGE_CHECK	<p>AXI インターコネクト コアがトランザクション エラー状態を検出するかどうかを次のように指定します。</p> <ul style="list-style-type: none"> ON (1) の場合、SI スロットで受信されたトランザクションに対するエラー状態をチェックします。その結果、AXI インターコネクト コアは BRESP 信号または RRESP 信号のいずれかによって、DECERR コードで応答する場合があります。 OFF (0) の場合、DECERR 応答を生成するロジックが AXI インターコネクトのインプリメンテーションから省略され、リソースが節約されます。

次の場合、C_RANGE_CHECK はデフォルトで ON (1) に設定されます。

- 複数の MI スロットがある
- MI スロットのいずれかに複数のアドレス範囲が定義されている
- AXI4-Lite の MI スロット (スレーブに接続) と AXI4-Lite 以外の SI スロット (マスターに接続) がある
- C_M_AXI_SECURE パラメーターがいずれかの MI スロットに設定されている

上記のデフォルト条件により、SI で受信されたプロトコル準拠のトランザクションは、プロトコルに準拠した完了コードか、正しい DECERR 応答のいずれかを必ず返すようになります。上記のどの条件も真でない場合、DECERR ロジックは (デフォルトで) 省略され、ロジックが削減されます。これにより、AXI インターコネクトをワイヤとしてインプリメントする「パススルー」でコンフィギュレーションすれば、リソースは使用されません。

接続されたマスターが通常 DECERR 応答のトリガーとなる特定のトランザクションを発行しないことが確実な場合には、C_RANGE_CHECK パラメーターを強制的に OFF にしてロジック リソースを節約できます。この条件を満足するのは、次のすべてが真であるときです。

- 存在しないスレーブにアクセスするマスターがない。これは、次のいずれかが真の場合に真になります。
 - アドレス範囲にかかわらず、(スレーブに接続される) MI スロットが 1 つしかない
 - すべてのトランザクションが、スパース クロスバーのコンネクティビティ マップと読み出し専用/書き込み専用のスレーブを含む設定済みアドレス マップの範囲内にあるアドレスを持つ
 - C_AXI_ADDR_WIDTH で定義されたとおり、アドレス マップがアドレス空間全体を包含している
- 次の不正トランザクションのいずれかによって AXI4-Lite スレーブにアクセスするマスターがない。
 - トランザクションの長さが 1 データ ビートよりも大きい
 - データ転送サイズが 4 バイトよりも大きい
- マスターの動作にかかわらず、MI スロットが、C_M_AXI_SECURE をイネーブルにしてコンフィギュレーションされていない。MI スロットのいずれかが SECURE としてコンフィギュレーションされている場合に C_RANGE_CHECK を OFF にしないでください。コンパイル時エラーが発生します。

MI スロットが 1 つしかなく、そのアドレス範囲が 1 つだけの場合、C_RANGE_CHECK パラメーターを強制的に ON にして無効なトランザクション アドレスをトラップできます。

次の条件は AXI インターコネクト コアでは検出されないため、C_RANGE_CHECK の影響を受けません。

- 認識できない応答 ID エラー
- AXI4 プロトコル違反 (AXI3 のアトミックな LOCK トランザクションを含む)
- 書き込みデータのインターリーブ
- 幅の狭いバースト違反 (C_S_AXI_SUPPORTS_NARROW_BURST を参照)

- 非同期クロック コンフィギュレーション以外でクロック倍率が整数でない
- パラメーター値範囲違反
- アドレスまたは ID の範囲が重複している、非バイナリ サイズである、またはベース値のアライメントが正しくない

パラメーター規則のまとめ

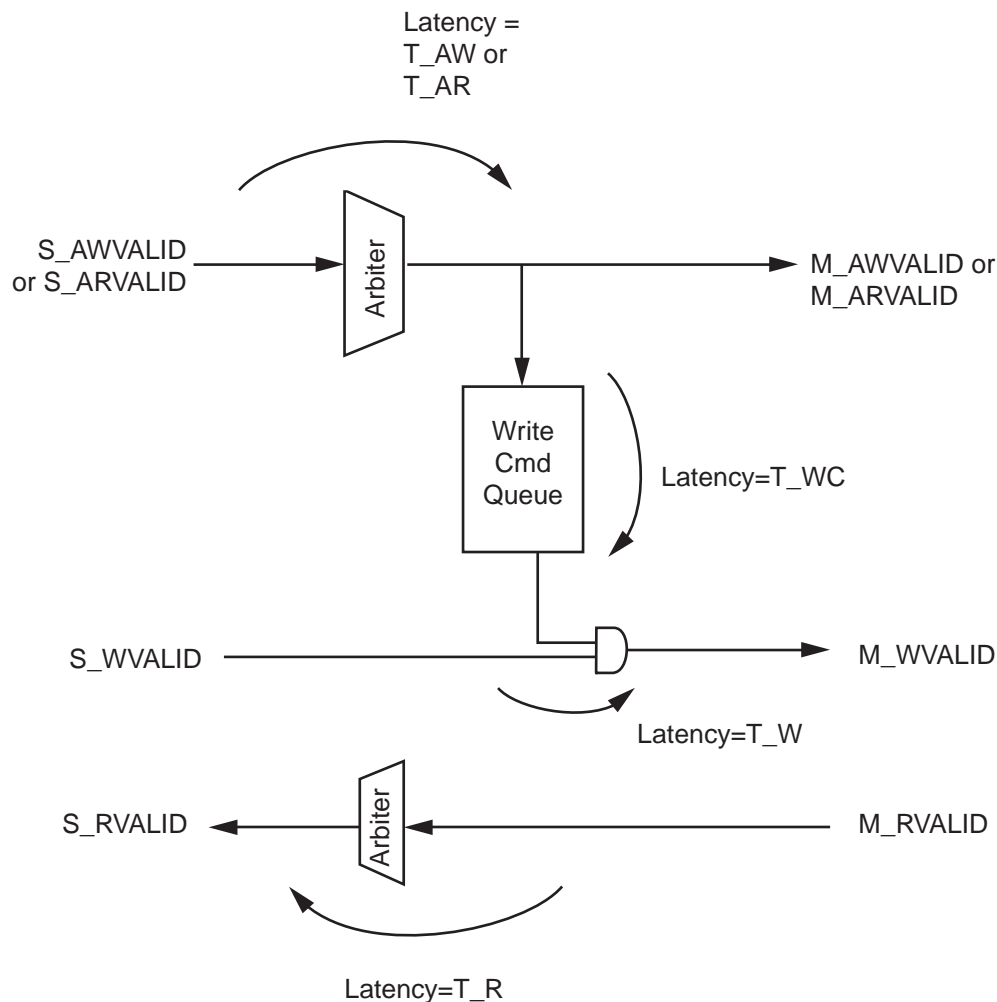
パラメーター表で説明した値の範囲のほかに、次の仕様規則が適用されます。

- いずれかの SI が 1024 ビット幅の場合、C_INTERCONNECT_DATA_WIDTH は 32 より大きい値に設定します。MI が 32 ビット幅の場合、C_INTERCONNECT_DATA_WIDTH は 1024 より小さい値に設定します。
- C_S_AXI_PROTOCOL が AXI4-Lite を示す場合、各スロットの C_S_AXI_DATA_WIDTH は 32 でなければなりません。C_M_AXI_PROTOCOL が AXI4-Lite を示す場合、それぞれの C_M_AXI_DATA_WIDTH は 32 でなければなりません。
- 使用されている各 MI スロットには、少なくとも 1 つのアドレス範囲が定義されている必要があります (null でないこと)。
- 各アドレス範囲では、範囲サイズが 2 のべき乗で、4096 以上でなければなりません。BASE_ADDR は範囲サイズの倍数でなければなりません (範囲内の位置を選択するすべての下位ビットは 0)。HIGH_ADDR の (範囲内の) 下位ビットはすべて 1 でなければなりません。
- 全 MI スロットのアドレス範囲のいずれかに重複があってはなりません。
- 使用されている各 SI スロットには、パラメーター C_S_AXI_BASE_ID および C_S_AXI_THREAD_ID_WIDTH を定義する必要があります。
- AXI4-Lite としてコンフィギュレーションされている SI スロットでは、C_S_AXI_IS_INTERCONNECT が 0 (エンドポイント マスター デバイス)、かつ C_S_AXI_THREAD_ID_WIDTH が 0 でなければなりません (AXI4-Lite インターフェイスでは ID 信号が使用されない)。
- 各 ID 範囲で、BASE_ID パラメーターのすべての下位ビット (THREAD_ID_WIDTH によって定義されたとおり) は、存在する場合、0 でなければなりません。言い換えると、BASE_ID は必ず範囲サイズの倍数です ($2 \times \text{THREAD_ID_WIDTH}$)。
- 全 SI スロットの ID 範囲のいずれかに重複があってはなりません。
- 各 ID 範囲の上限 ($\text{BASE_ID} + 2 \times \text{THREAD_ID_WIDTH} - 1$) は AXI インターコネクトの ID の最大値 ($2 \times \text{C_AXI_ID_WIDTH} - 1$) を超えることはできません。
- 各 SI スロットおよび各 MI スロットについて、SUPPORTS_WRITE または SUPPORTS_READ (もしくはその両方) が有効の必要があります。

パフォーマンス

レイテンシ

図 12 に、クロスバー モジュールのレイテンシ モデルのベースライン (バイパス モードでない状態) を示します。



X12091

図 12: クロスバー モデルのベースライン レイテンシ

図 12 において、ベースライン レイテンシは次のように表されます。

- $AW/ARVALID$ の順方向の伝搬では、 $T_AW = T_AR = 2 \text{ INTERCONNECT_ACLK}$ サイクルになります。ただし、アービトレーションにより使用权を得る妨げとなる保留状態 (より優先順位の高い要求があるなど) がない場合に限りです。また、アービトレーションのたびに 2 バブル サイクルが生じるため、同じ SI スロットによる連続したアービトレーションの間隔は 3 サイクル (最小値) になります。
- $T_WC = 1 \text{ INTERCONNECT_ACLK}$ サイクルです。
- 各 MI スロットで $C_M_AXI_PROTOCOL \neq AXI4Lite$ 、かつ $C_M_AXI_DATA_WIDTH \neq C_INTERCONNECT_DATA_WIDTH$ の場合、 $T_W = 1 \text{ INTERCONNECT_ACLK}$ サイクルとなり、バブル サイクルは発生しません (back-to-back の連続データ転送をサポート)。それ以外の場合、 $T_W = 0$ サイクルとなります (経由するクロスバーと組み合わせた値)。

- $T_R = 1$ または 2 INTERCONNECT_ACLK サイクル。バブル サイクルなし (連続する back-to-back データ転送をサポート)。2 回目のレイテンシ サイクルはアイドル サイクル後に再アービトレーションが行われる (要求 MI スロットが最後に使用権を与えられた MI スロットと異なる) 場合に発生します。同じ MI スロットが back-to-back でデータを伝搬する間、または複数の MI スロットが連続してデータをインターリーブしている間は、R チャネル アービタのレイテンシは 1 サイクルです。
- T_B (B チャネル レイテンシ、図には記載なし) $= 1$ または 2 INTERCONNECT_ACLK サイクル (T_R と同様) です。

次に挙げるようなクロスバーの外側のさまざまなオプション モジュールにより、レイテンシ サイクルが増加します。

- 「FULLY_REGISTERED」レジスタ スライス (適用可能な各チャネル): S_AXI_ACLK または M_AXI_ACLK の 1 レイテンシ サイクル、バブル サイクルなし (チャネル帯域幅 100% の最良条件)
- 「LIGHT_WEIGHT」レジスタ スライス (適用可能な各チャネル): S_AXI_ACLK または M_AXI_ACLK の 1 レイテンシ サイクル、 1 バブル サイクル (チャネル帯域幅 50% の最良条件)、AW、AR、および B チャネル転送と AXI4-Lite エンドポイントを使用するすべての転送に適する
- データ FIFO
 - W チャネルと R チャネル: INTERCONNECT_ACLK の 3 レイテンシ サイクル、バブル サイクルなし
 - AW、AR、および B チャネル: レイテンシなし
- クロック変換: レイテンシは変動
- アップサイザー
 - AW チャネルと AR チャネル: 1 レイテンシ サイクル
 - W チャネル: 1 レイテンシ サイクル (圧縮が完了するサイクルごと)、SI 側の (狭い) インターフェイスにはバブル サイクルなし
 - R チャネル: 1 レイテンシ サイクル
 - B チャネル: レイテンシなし
 - クロック
 - SI 域のアップサイザーは INTERCONNECT_ACLK によってクロック供給される
 - MI 域のアップサイザーは M_AXI_ACLK によってクロック供給される
- ダウンサイザー
 - AW チャネルと AR チャネル: 1 レイテンシ サイクル
 - R チャネル: レイテンシなし (圧縮が完了するサイクルごと)、MI 側の (狭い) インターフェイスにはバブル サイクルなし
 - W チャネル: レイテンシなし
 - B チャネル: レイテンシなし
 - クロック
 - SI 域のダウンサイザーは S_AXI_ACLK によってクロック供給される
 - MI 域のダウンサイザーは INTERCONNECT_ACLK によってクロック供給される
- AXI4-Lite 変換: いずれのチャネルにもレイテンシなし
- AXI3 変換
 - AW チャネルと AR チャネル: M_AXI_ACLK の 1 レイテンシ サイクル
 - W、R、および B チャネル: レイテンシなし

クロック周波数

このセクションでは、Spartan-6 および Virtex-6 ファミリーにおいて期待される AXI インターコネクトのクロック周波数について説明します。実際に実現可能なクロック周波数は、FPGA のロジック使用率、物理的配置制約 (I/O など)、ツール オプション、およびその他の要因によって変動します。

AXI インターコネクトでダウンサイザー機能を使用すると、タイミングがターゲットとする Fmax 値がそれ以上また h 以下となることがあります。

Nx1 コンフィギュレーション

次のコンフィギュレーション オプションは、複数のマスター デバイスから 1 つの高性能なスレーブ デバイス (メモリ コントローラーなど) へのアクセスを提供するアプリケーションを示します。

- 最大 4 つの SI スロットおよび 1 つの MI スロット
- MI データ幅 64 ビット
- 内部データ幅 64 ビット
- 32 ビット幅の SI スロットを処理するアップサイザー (必要な場合)
- クロック変換モジュール (必要な場合)
- データパス FIFO (必要な場合)

上記のアプリケーションで実現可能な AXI インターコネクトのシステム内クロック周波数は、次のとおりです。

- -1 スピード グレードの Virtex-6 デバイス : 200MHz
- -3 スピード グレードの Spartan-6 デバイス : 133MHz

その他のコンフィギュレーション

最大 8 つの SI スロットを持つ NxM クロスバーなどの大規模なコンフィギュレーションでは、通常、次のクロック周波数をサポートします。

- -1 スピード グレードの Virtex-6 FPGA : 167MHz
- -3 スピード グレードの Spartan-6 FPGA : 111MHz

リソース使用率

表 13 および表 14 に、AXI インターコネクト コア内部のさまざまなモジュールの推定 FPGA リソース使用率を示します。これらには、各モジュールの一般的なコンフィギュレーションも列記されています。AXI インターコネクトの特定インスタンスの総エリアは、構成モジュールすべての使用率を累積することで推定できます。

表 13 : Virtex-6 FPGA リソース使用率

モジュール	コンフィギュレーション	FF	LUT	ブロック RAM
クロスバー (SAMD)	4 SI x 1 MI、データ幅 = 32 ビット	300	740	
クロスバー (SAMD)	4 SI x 1 MI、データ幅 = 64 ビット	390	940	
クロスバー (SAMD)	1 SI x 4 MI、データ幅 = 32 ビット	340	400	
共有アクセス クロスバー (SASD)	1 SI x 4 MI、データ幅 = 32 ビット	35	150	
クロスバー (SAMD)	4 SI x 4 MI、データ幅 = 32 ビット	560	1530	
アップサイザー	データ幅 = 32 ~ 64 ビット	420	560	
ダウンサイザー	データ幅 = 64 ~ 32 ビット	560	620	
同期クロック コンバーター	データ幅 = 32 ビット	180	70	

表 13 : Virtex-6 FPGA リソース使用率 (続き)

モジュール	コンフィギュレーション	FF	LUT	ブロック RAM
同期クロック コンバーター	データ幅 = 64 ビット	330	110	
非同期クロック コンバーター	データ幅 = 32 ビット	720	350	
非同期クロック コンバーター	データ幅 = 64 ビット	870	400	
データパス FIFO	深さ 32 (LUT RAM)、データ幅 = 32 ビット	110	60	
データパス FIFO	深さ 32 (LUT RAM)、データ幅 = 64 ビット	180	90	
データパス FIFO	深さ 512 (ブロック RAM)、データ幅 = 32 ビット	90	60	1
データパス FIFO	深さ 512 (ブロック RAM)、データ幅 = 64 ビット	130	60	2
レジスタ スライス	AW/AR チャンネル、軽量化 (タイプ 7)	30	5	
レジスタ スライス	B チャンネル、軽量化 (タイプ 7)	5	5	
レジスタ スライス	W/R チャンネル、軽量化 (タイプ 7)、データ幅 = 32 ビット	40	10	
レジスタ スライス	W/R チャンネル、軽量化 (タイプ 7)、データ幅 = 64 ビット	80	10	
レジスタ スライス	W/R チャンネル、完全レジスタ化 (タイプ 1)、データ幅 = 32 ビット	80	50	
レジスタ スライス	W/R チャンネル、完全レジスタ化 (タイプ 1)、データ幅 = 64 ビット	150	80	
AXI4 から AXI3 へのコンバーター	データ幅 = 32 ビット	240	250	
AXI4 から AXI4-Lite へのコンバーター	データ幅 = 32 ビット	5	15	

表 14 : Spartan-6 FPGA リソース使用率

モジュール	コンフィギュレーション	FF	LUT	ブロック RAM
クロスバー (SAMD)	4 SI x 1 MI、データ幅 = 32 ビット	300	740	
クロスバー (SAMD)	4 SI x 1 MI、データ幅 = 64 ビット	390	970	
クロスバー (SAMD)	1 SI x 4 MI、データ幅 = 32 ビット	340	400	
共有アクセス クロスバー (SASD)	1 SI x 4 MI、データ幅 = 32 ビット	40	150	
クロスバー (SAMD)	4 SI x 4 MI、データ幅 = 32 ビット	560	1430	
アップサイザー	データ幅 = 32 ~ 64 ビット	420	560	
ダウンサイザー	データ幅 = 64 ~ 32 ビット	560	630	
同期クロック コンバーター	データ幅 = 32 ビット	190	75	
同期クロック コンバーター	データ幅 = 64 ビット	330	120	
非同期クロック コンバーター	データ幅 = 32 ビット	720	330	
非同期クロック コンバーター	データ幅 = 64 ビット	870	400	
データパス FIFO	深さ 32 (LUT RAM)、データ幅 = 32 ビット	110	50	
データパス FIFO	深さ 32 (LUT RAM)、データ幅 = 64 ビット	180	70	
データパス FIFO	深さ 512 (ブロック RAM)、データ幅 = 32 ビット	90	50	2

表 14 : Spartan-6 FPGA リソース使用率 (続き)

モジュール	コンフィギュレーション	FF	LUT	ブロック RAM
データパス FIFO	深さ 512 (ブロック RAM)、データ幅 = 64 ビット	130	70	3
レジスタ スライス	AW/AR チャンネル、軽量化 (タイプ 7)	30	5	
レジスタ スライス	B チャンネル、軽量化 (タイプ 7)	5	5	
レジスタ スライス	W/R チャンネル、軽量化 (タイプ 7)、データ幅 = 32 ビット	40	10	
レジスタ スライス	W/R チャンネル、軽量化 (タイプ 7)、データ幅 = 64 ビット	80	10	
レジスタ スライス	W/R チャンネル、完全レジスタ化 (タイプ 1)、データ幅 = 32 ビット	80	50	
レジスタ スライス	W/R チャンネル、完全レジスタ化 (タイプ 1)、データ幅 = 64 ビット	150	80	
AXI4 から AXI3 へのコンバーター	データ幅 = 32 ビット	200	180	
AXI4 から AXI4-Lite へのコンバーター	データ幅 = 32 ビット	5	15	

サポート

ザイリンクスは、製品資料の説明に従って使用されている LogiCORE IP 製品に対するテクニカル サポートを提供しています。製品資料で定義されていないデバイスにインプリメントしたり、許容されている範囲を超えてカスタマイズしたり、「DO NOT MODIFY」とされているデザイン セクションに変更を加えたりした場合、タイミング、機能、製品サポートは保証されません。

注文情報

このザイリンクス LogiCORE IP モジュールは、ザイリンクス ISE Design Suite Embedded Edition ソフトウェアに無償で含まれ、ザイリンクスの[エンド ユーザー ライセンス](#)の条件の下で使用できます。コアは、ザイリンクス ISE Embedded Edition ソフトウェア (EDK) によって生成されます。このモジュールに関する詳細は、[AXI インターコネクト](#)のページに記載されています。

その他のザイリンクス LogiCORE IP モジュールに関する情報は[ザイリンクス IP](#)のページをご覧ください。その他のザイリンクス LogiCORE モジュールおよびソフトウェアの価格および提供状況については、お近くのザイリンクス[販売代理店](#)にお問い合わせください。

参考資料

- 『ARM AMBA AXI プロトコル v2.0』(文書番号 ARM IHI 0022C)
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ih0022c/index.html>

次の資料は、[ザイリンクスのサポート ウェブサイト](#)から入手できます。

- 『Platform Specification Format Reference Manual』(UG642)
- 『ザイリンクス AXI リファレンス ガイド』(UG761)
- 『LogiCORE IP AXI-to-AXI Connector』(DS803)

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂の説明
2010/09/21	1.0	ザイリンクスによる初回 IDS リリース 12.3。
2010/12/14	2.0	ザイリンクスによる IDS リリース 12.4。
2011/03/01	3.0	ザイリンクスによる IDS リリース 13.1。
2011/06/22	4.0	ザイリンクスによる IDS リリース 13.2。
2011/10/19	4.1	<p>コア バージョン 1.04.a 対応の ISE 13.3 ソフトウェアのリリース</p> <ul style="list-style-type: none"> • C_INTERCONNECT_DATA_WIDTH パラメーターの説明から SI スロットを削除。 • 2 ページの「XPS でサポートされる機能」から、AXI4/AXI3 バーストが変換されないことを説明する箇条書きの項目を追加。 • 「CORE Generator ツールでサポートされる機能」から、サポートされるスレーブとマスターのインターフェイス数に関する箇条書きの項目を変更。 • 22 ページの「ID 信号の使用法」に、M_AXI_WID と S_AXI_WID 信号に関するパラグラフを追加。 • 27 ページの「AXI4-Lite スレーブ変換」に、メモを追加。 • 表 3 に S_AXI_WID とメモ 4 を追加。 • 表 4 に、メモ 2 を追加。 • 表 7 に、C_S_AXI_WRITE_FIFO_DELAY と C_S_AXI_READ_FIFO_DELAY を追加。 • 表 8 に、C_M_AXI_WRITE_FIFO_DELAY と C_M_AXI_READ_FIFO_DELAY を追加。 • 表 10 に、C_Snn_AXI_WRITE_FIFO_DELAY と C_Snn_AXI_READ_FIFO_DELAY を追加。 • 表 11 に、C_M00_AXI_WRITE_FIFO_DELAY と C_M00_AXI_READ_FIFO_DELAY を追加。 • 49 ページの「データパス FIFO」を改訂。
2012/01/18	5.0	<p>コア バージョン 1.05.a 対応の ISE 13.4 ソフトウェアのリリース</p> <ul style="list-style-type: none"> • 46 ページの「トランザクションの受け入れ限度と発行限度」の説明を明確化。 • 表 7、表 8、表 10、および表 11 に FIFO パケットの読み出しと書き込み動作についての説明を追加。

Notice of Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

本資料は英語版 (v5.0) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。