

AMBA[®] AXI プロトコル

v1.0

仕様書

ARM[®]

AMBA AXI プロトコル 仕様書

Copyright © 2003, 2004 ARM Limited. All rights reserved.

リリース情報

改訂履歴

日付	版	変更点
2003 年 6 月 16 日	A	初版リリース
2004 年 3 月 19 日	B	読み出しアドレスチャネルおよび書き込みアドレスチャネルの定義

著作権

® または ™ の付いた言葉とロゴは、本著作権条項で特に明記されていない限り、EU および他国における ARM 社の登録商標または商標です。本書に記載されている他の製品名は、各社の所有する商標です。

本書に記載されている情報の全部または一部、ならびに本書で紹介する製品は、著作権所有者の文書による事前の許可を得ない限り、転用・複製することを禁じます。

本書に記載されている製品は、今後も継続的に開発・改良の対象となります。本書に含まれる製品およびその利用方法についての情報は、ARM 社が利用者の利益のために提供するものです。したがって当社では、製品の商品性または目的への適用性を含め、暗黙的・明示的に関係なく一切の責任を負いません。

本書は、本製品の利用者をサポートすることだけを目的としています。本書に記載されている情報の使用、情報の誤りまたは省略、あるいは本製品の誤使用によって発生したいかなる損失・損害についても、ARM 社は一切責任を負いません。

AMBA 仕様の使用許諾

1. 条項 2 および 3 の規定に基づき、ARM は被使用許諾者に対し、AMBA 仕様準拠製品の開発、製造、販売、供給、あるいは配布を目的として AMBA 仕様を使用および複製する永久的かつ非独占的で、譲渡不可能な使用料無料の世界ライセンスを供与します。
2. AMBA 仕様は、「現状有姿」で提供されるものです。品質、商品性、特定の目的に対する適合性または第三者の権利を侵害しない旨の保証などを含め、ARM は本仕様に関し明示的または黙示的を問わず何らの保証も致しません。また本項はユーザの法律上の権利行使を制限するものではありません。
3. 条項 1 の規定の下、AMBA 仕様またはこれに基づくいかなる製品に関しても、被使用許諾者には明示的または黙示的を問わず、ARM の商標もしくは AMBA の商標を使用する権利は供与されません。条項 1 は、AMBA 仕様に関して ARM に代わって抗議する権限を被使用許諾者に付与するものではありません。

守秘義務

本書は誰でも閲覧することができます。配布に関する制限はありません。

製品ステータス

本書には最終情報（完成製品に関する情報）が記載されています。

ARM ホームページ

<http://www.arm.com>

目次

AMBA AXI プロトコル仕様書

序章

本書について	xiv
ご意見	xviii

第 1 章

はじめに

1.1	AXI プロトコルについて	1-2
1.2	アーキテクチャ	1-3
1.3	基本トランザクション	1-7
1.4	その他の機能	1-11

第 2 章

シグナルの説明

2.1	グローバルシグナル.....	2-2
2.2	書き込みアドレスチャネルシグナル	2-3
2.3	書き込みデータチャネルシグナル	2-4
2.4	書き込み応答チャネルシグナル.....	2-5
2.5	読み出しアドレスチャネルシグナル	2-6
2.6	読み出しデータチャネルシグナル	2-7
2.7	低電力インタフェースシグナル.....	2-8

第 3 章	チャネルハンドシェイク	
	3.1 ハンドシェイクプロセス	3-2
	3.2 チャネル間の関係	3-6
	3.3 チャネルハンドシェイクシグナル間の依存関係	3-7
第 4 章	アドレッシングオプション	
	4.1 アドレッシングオプションについて	4-2
	4.2 バースト長	4-3
	4.3 バーストサイズ	4-4
	4.4 バーストタイプ	4-5
	4.5 バーストアドレス	4-7
第 5 章	その他の制御情報	
	5.1 キャッシュのサポート	5-2
	5.2 保護ユニットのサポート	5-5
第 6 章	アトミックアクセス	
	6.1 アトミックアクセスについて	6-2
	6.2 排他アクセス	6-3
	6.3 ロックアクセス	6-7
第 7 章	応答シグナル	
	7.1 応答シグナルについて	7-2
	7.2 応答のタイプ	7-4
第 8 章	順序モデル	
	8.1 順序モデルについて	8-2
	8.2 転送 ID フィールド	8-3
	8.3 読み出しの順序	8-4
	8.4 通常書き込みの順序	8-5
	8.5 書き込みデータのインタリーブ	8-6
	8.6 読み出しと書き込みの相互作用	8-8
	8.7 相互接続による ID フィールドの使用	8-9
	8.8 推奨される ID フィールドの幅	8-10
第 9 章	データバス	
	9.1 データバスについて	9-2
	9.2 書き込みストローブ	9-3
	9.3 ナロー転送	9-4
	9.4 バイト不変	9-5
第 10 章	アンアラインド転送	
	10.1 アンアラインド転送について	10-2
	10.2 例	10-3

第 11 章	クロックとリセット	
	11.1 クロックおよびリセットの要件.....	11-2
第 12 章	低電力インタフェース	
	12.1 低電力インタフェースについて	12-2
	12.2 低電力クロック制御.....	12-3

表一覧

AMBA AXI プロトコル仕様書

	改訂履歴	ii
表 2-1	グローバルシグナル	2-2
表 2-2	書き込みアドレスチャネルシグナル	2-3
表 2-3	書き込みデータチャネルシグナル	2-4
表 2-4	書き込み応答チャネルシグナル	2-5
表 2-5	読み出しアドレスチャネルシグナル	2-6
表 2-6	読み出しデータチャネルシグナル	2-7
表 2-7	低電力インタフェースシグナル	2-8
表 4-1	バースト長のエンコーディング	4-3
表 4-2	バーストサイズのエンコーディング	4-4
表 4-3	バーストタイプのエンコーディング	4-5
表 5-1	キャッシュのエンコーディング	5-3
表 5-2	保護レベルのエンコーディング	5-6
表 6-1	アトミックアクセスのエンコーディング	6-2
表 7-1	RRESP[1:0] シグナルと BRESP[1:0] シグナルのエンコーディング	7-2

図一覽

AMBA AXI プロトコル仕様書

図 1-1	読み出しのチャネルアーキテクチャ	1-3
図 1-2	書き込みのチャネルアーキテクチャ	1-4
図 1-3	インタフェースと相互接続	1-5
図 1-4	読み出しバースト	1-7
図 1-5	読み出しバーストのオーバーラップ	1-8
図 1-6	書き込みバースト	1-9
図 3-1	VALID の後に READY が通知されるハンドシェイク	3-2
図 3-2	READY の後に VALID が通知されるハンドシェイク	3-3
図 3-3	VALID と READY が同時に通知されるハンドシェイク	3-3
図 3-4	読み出しトランザクションにおけるハンドシェイクの依存関係	3-7
図 3-5	書き込みトランザクションにおけるハンドシェイクの依存関係	3-8
図 9-1	バイトレーンのマッピング	9-3
図 9-2	各転送サイズが 8 ビットのナロー転送	9-4
図 9-3	各転送サイズが 32 ビットのナロー転送	9-4
図 9-4	エンディアン混在形式のデータ構造の例	9-5
図 10-1	32 ビットバス上のアラインドおよびアンアラインドワード転送	10-3
図 10-2	64 ビットバス上のアラインドおよびアンアラインドワード転送	10-4
図 10-3	64 ビットバス上のアラインドラッピングワード転送	10-4
図 11-1	リセット後の終了	11-2
図 12-1	CSYSREQ と CSYSACK のハンドシェイク	12-3
図 12-2	低電力要求の受け入れ	12-4
図 12-3	低電力要求の拒否	12-5
図 12-4	低電力クロック制御シーケンス	12-6

序章

本章では、*AMBA* アドバンストエクステンシブルインタフェース (*AXI*) プロトコル仕様とその関連資料について説明します。本章は以下のセクションから構成されています。

- 本書について : P. xiv
- ご意見 : P. xviii

本書について

本書は、AMBA AXI プロトコル仕様書 v1.0 です。前の仕様書 r0p0 版の新しいバージョンです。

対象となる読者

本書は、アドバンストマイクロコントローラバスアーキテクチャ（AMBA）を熟知し、AXIに準拠するシステムおよびモジュールの設計に携わるハードウェア/ソフトウェアエンジニアをサポートする目的で書かれています。

構成

本書は以下の章から構成されています。

第1章 「はじめに」

この章では、AXI プロトコルのアーキテクチャと、AXI プロトコルで定義されている基本的なトランザクションについて説明します。

第2章 「シグナルの説明」

この章では、AXI が備えるグローバルな、書き込みアドレスチャネル、書き込みデータチャネル、書き込み応答チャネル、読み出しアドレスチャネル、読み出しデータチャネル、低電力インタフェースシグナルについて説明します。

第3章 「チャネルハンドシェイク」

AXI チャネルハンドシェイクプロセスについて説明します。

第4章 「アドレッシングオプション」

AXI のバーストタイプと、バースト転送に使用されるアドレスおよびバイトレーンの計算方法について説明します。

第5章 「その他の制御情報」

システムレベルのキャッシュおよび保護ユニットをサポートする場合の、AXI プロトコルの使用方法について説明します。

第6章 「アトミックアクセス」

排他アクセスおよびロックアクセスの実行方法について説明します。

第7章 「応答シグナル」

AXI スレーブの4つのトランザクション応答について説明します。

第 8 章 「順序モデル」

Out-of-Order トランザクションの処理を可能にするため、AXI によってどのようにトランザクション ID タグが使用されるかについて説明します。

第 9 章 「データベース」

AXI の読み出しデータベースと書き込みデータベースにおけるサイズの異なるトランザクションの処理方法と、バイト不変エンディアンを使用したエンディアン混在データの処理方法について説明します。

第 10 章 「アンアラインド転送」

AXI プロトコルによる非境界整列転送の処理方法について説明します。

第 11 章 「クロックとリセット」

AXI のクロックシグナルとリセットシグナルのタイミングについて説明します。

第 12 章 「低電力インタフェース」

AXI クロック制御インタフェースを使用した低電力状態の開始 / 終了方法について説明します。

規則

本仕様書で使用されている表記規則については、以下のセクションで説明します。

- 表記表記
- タイミング図 : P. xvi
- シグナル : P. xvii

表記表記

本仕様書では、以下の表記規則を用いています。

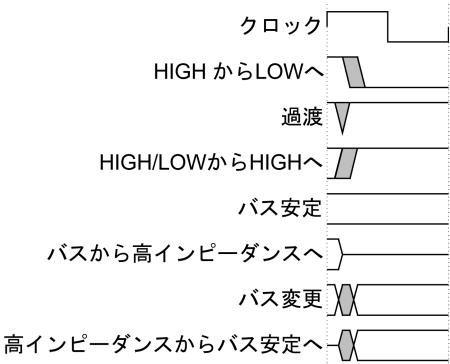
<i>italic</i>	重要事項、重要用語、相互参照、引用箇所を斜体で記載しています。
bold	メニュー名やボタンなどのインタフェース要素を太字で記載しています。また、必要に応じて表中の説明文に含まれる専門用語にも太字を用いています。
monospace	コマンド、ファイル名、プログラム名、ソースコードなどの、キーボードから入力可能なテキストを示しています。

<code>monospace</code>	コマンドまたはオプションに使用可能な略語を示しています。コマンドやオプションの名前を全部入力する代わりに、下線部分のテキストだけを入力してこれらを指定できます。
<code>monospace italic</code>	コマンドまたは関数への引数で、特定の値に置き換えられることが可能なものをタイプライター書体の太字で記載しています。
<code>monospace bold</code>	サンプルコード以外で使用されている場合は、言語のキーワードと ARM プロセッサのシグナル名をタイプライター書体の太字で記載しています。
<code>< and ></code>	コードまたはコード片の中で鍵括弧が使用されている場合は、その鍵括弧で囲まれた言葉がアセンブラ構文内で置き換え可能なことを示しています。文章の中では以下のように通常のフォントで記載されています。例： <ul style="list-style-type: none">MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>Opcode_2 の値によって、アクセスされるレジスタが選択されます。

タイミング図

タイミング図の規則を表す記号は、これらのタイミング図に使用されている構成要素を示しています。この図と異なる意味で使用されている場合は、その都度明記しています。タイミング図に明記されていないタイミング情報を推測で判断しないようにして下さい。

影付きのバス領域とシグナル領域は定義されていません。そのため、影付きの領域内では、バスおよびシグナルは任意の値を取り得ます。実際のレベルは重要ではなく、通常の動作に影響を及ぼすことはありません。



タイミング図の規則を表す記号

シグナル

シグナルの表記規則は以下の通りです。

シグナルのレベル	アサートされているシグナルのレベルはハイでアクティブかローでアクティブかによって異なります。アサートされている状態は、ハイでアクティブなシグナルではハイ、ローでアクティブなシグナルではローになります。
接頭辞 A	AXI のグローバルな書き込みアドレスチャネルシグナルと読み出しアドレスチャネルシグナルを示します。
接頭辞 B	AXI の書き込み応答チャネルシグナルを示します。
接頭辞 C	AXI の低電力インタフェースシグナルを示します。
接頭辞 R	AXI の読み出しデータチャネルシグナルを示します。
接頭辞 W	AXI の書き込みデータチャネルシグナルを示します。
接尾辞 n	AXI の ARESETn グローバルリセットシグナルがローでアクティブであることを示します。

参考資料

このセクションでは、ARM 社から提供されている出版物を紹介します。

ARM は資料の定期的な更新・修正を行っています。以下の情報を参照して下さい。
最新の正誤表と追補情報、ならびに ARM と ARBA に関するよく寄せられる質問とその回答については、<http://www.arm.com> および <http://www.amba.com> をご覧下さい。

ARM の出版物

本書では AXI プロトコルに限定した情報を提供しています。他の関連情報については、以下の出版物をご覧下さい。

- *AMBA 仕様書 (Rev 2.0)* (ARM IHI 0011)
- *ARM アーキテクチャリファレンスマニュアル* (ARM DDI 0100)

ご意見

ARM 社では、AMBA AXI プロトコルおよび本書に関するご意見をお待ちしています。

本製品に関するご意見

AXI に関するご意見がございましたら、以下の情報とともに製品購入元までご連絡下さい。

- 製品名
- 問題点の詳しい説明

本書に関するご意見

本書に関するご意見等がございましたら、電子メールに以下の情報をご記入の上、errata@arm.com までお寄せ下さい。

- 資料名
- 資料番号
- ご意見のあるページ番号
- ご意見の詳しい説明

ARM 社では、追記や改善に関するお客様からのご要望をお待ちしております。

第 1 章

はじめに

本章では、AXI プロトコルのアーキテクチャと、AXI プロトコルによって定義される基本トランザクションについて説明します。本章は以下のセクションから構成されています。

- AXI プロトコルについて : P. 1-2
- アーキテクチャ : P. 1-3
- 基本トランザクション : P. 1-7
- その他の機能 : P. 1-11

1.1 AXI プロトコルについて

AMBA AXI プロトコルは、高性能な高周波数のシステムを対象としており、高速なサブミクロンレベルの相互接続に適応するための数多くの機能を備えています。

この新世代 AMBA インタフェースの目的は以下の通りです。

- 高帯域幅・低レイテンシの設計に適応する。
- 複雑なブリッジを使用せずに高周波数での動作を実現する。
- 広範なコンポーネントのインタフェース要件を満たす。
- 初回アクセスのレイテンシが高いメモリコントローラに適応する。
- 相互接続アーキテクチャの実装に柔軟性を持たせる。
- 既存の AHB および APB インタフェースへの下位互換性を維持する。

AXI プロトコルの主な特徴は以下の通りです。

- アドレス / 制御フェーズとデータフェーズが独立している。
- バイトストロープを使用した非境界整列データ転送がサポートされている。
- 開始アドレスの発行のみでバースト指向のトランザクションが行われる。
- 読み出しデータチャネルと書き込みデータチャネルが独立していることで、低コストのダイレクトメモリアクセス (DMA) が実現可能
- 複数のアドレスの発行が可能
- Out-of-Order トランザクションが実現されている。
- タイミングクロージャを提供するためのレジスタステージの追加が容易

基本データ転送プロトコルに加えて、AXI プロトコルには低電力動作に合わせたシグナル生成を可能にするオプションの拡張機能があります。

1.2 アーキテクチャ

AXI プロトコルはバースト指向のプロトコルです。各トランザクションには、転送されるデータの性質を表すアドレスチャンネルに関するアドレス情報と制御情報が保持されています。データは、スレーブへの書き込みチャンネル、またはマスタへの読み出しチャンネルを使用して、マスタとスレーブとの間で転送されます。AXI プロトコルでは、データがすべてマスタからスレーブへ送られる書き込みトランザクションの際に、別の書き込み応答チャンネルを使用して、スレーブが書き込みトランザクションの完了をマスタに通知することができます。

AXI プロトコルでは、以下のような処理が可能です。

- 実際のデータ転送の前にアドレス情報を発行すること。
- 複数のトランザクション発行
- トランザクションの Out-of-Order での完了をサポートすること。

図 1-1 は、読み出しトランザクションにおける読み出しアドレスチャンネルと読み出しデータチャンネルの使用方法を示しています。

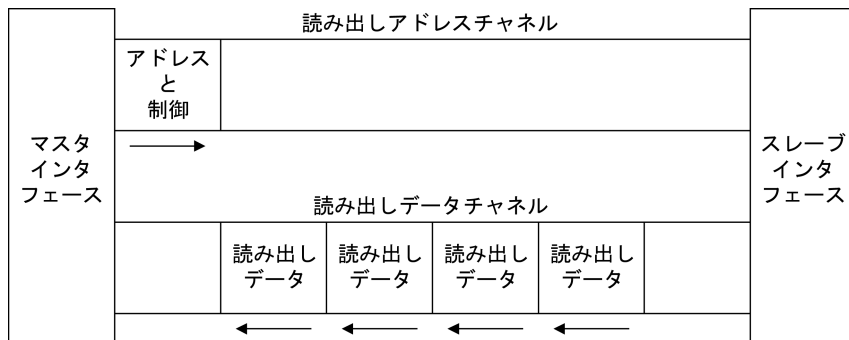


図 1-1 読み出しのチャンネルアーキテクチャ

P. 1-4 図 1-2 は、書き込みトランザクションにおける書き込みアドレスチャンネル、書き込みデータチャンネル、書き込み応答チャンネルの使用方法を示しています。

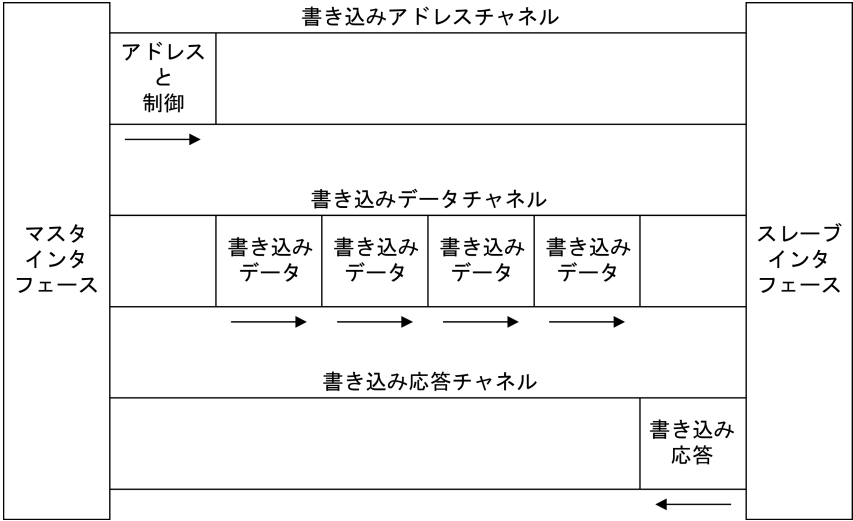


図 1-2 書き込みのチャンネルアーキテクチャ

1.2.1 チャンネルの定義

5 つの独立したチャンネルはそれぞれ情報シグナル群から構成され、**VALID** と **READY** の 2 ウェイハンドシェイクメカニズムが使用されます。

情報ソースは **VALID** シグナルを使用して、有効なデータと制御情報がチャンネル上で利用可能であることを通知します。デスティネーションはそのデータを受け取る準備ができると、**READY** シグナルを使用してこれを通知します。また、読み出しデータチャンネルと書き込みデータチャンネルには、トランザクション内で最後のデータが送信されたことを示す **LAST** シグナルがあります。

読み出しアドレスチャンネルと書き込みアドレスチャンネル

読み出しトランザクションと書き込みトランザクションにはそれぞれ独自のアドレスチャンネルがあります。それぞれのアドレスチャンネルによって、トランザクションに必要なアドレス情報と制御情報が伝達されます。AXI プロトコルは以下のメカニズムをサポートしています。

- 1 バーストにつき 1 ～ 16 データ転送の可変長バースト
- 8 ～ 1024 ビットの転送サイズによるバースト
- ラッピングバースト、インクリメントバースト、非インクリメントバースト
- 排他 / ロックアクセスを使用したアトミック処理
- システムレベルのキャッシングおよびバッファリング制御

- セキュアな特権アクセス

読み出しデータチャンネル

読み出しデータチャンネルは、読み出しデータと、スレーブからマスタへの読み出し応答情報の両方を転送します。読み出しデータチャンネルは以下から構成されます。

- データ幅が 8、16、32、64、128、256、512、あるいは 1024 ビットのデータバス
- 読み出しトランザクションの完了ステータスを通知する読み出し応答

書き込みデータチャンネル

書き込みチャンネルはマスタからスレーブへの書き込みデータを転送します。このチャンネルは以下から構成されます。

- データ幅が 8、16、32、64、128、256、512、あるいは 1024 ビットのデータバス
- 8 データビット毎にデータバスのどのバイトが有効であることを示す、1 バイトレーンストローブ

書き込みチャンネルは常にバッファリングされるものとして扱われるため、前の書き込みトランザクションに対するスレーブからの応答がなくても、マスタは書き込みトランザクションを実行できます。

書き込み応答チャンネル

書き込み応答チャンネルにより、スレーブは書き込みトランザクションへの応答を返すことができます。どの書き込みトランザクションにも完了シグナルが使用されます。

完了シグナルは、バースト内の個々のデータ転送に対してではなく、各バーストに対して 1 度だけ生成されます。

1.2.2 インタフェースと相互接続

典型的なシステムは、図 1-3 に示すような何らかの形の相互接続によって接続された複数のマスタデバイスとスレーブデバイスで構成されます。

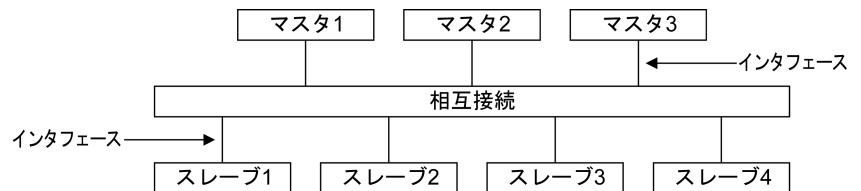


図 1-3 インタフェースと相互接続

AXI プロトコルによって、以下の接続に対して 1 つのインタフェースが定義されます。

- マスタと相互接続の間
- スレーブと相互接続の間
- マスタとスレーブの間

このインタフェースの定義により、様々な相互接続を実装できます。デバイス間の相互接続は、実際のマスタデバイスとスレーブデバイスを接続できる対称のマスタポートとスレーブポートをもつ別のデバイスに相当します。

ほとんどのシステムで、相互接続には以下のいずれかの方法が使用されます。

- 共有アドレスバスおよびデータバス
- 共有アドレスバスと多重データバス
- 多重アドレスバスと多重データバスをもつマルチレイヤ

ほとんどのシステムでは、データチャネルの帯域幅に比べ、アドレスチャネルの帯域幅の方がかなり少なくてすみます。このようなシステムでは、共有アドレスバスと多重データバスを使用して並列データ転送を可能にすることで、システムのパフォーマンスと相互接続の複雑度とのバランスを取ることができます。

1.2.3 レジスタスライス

各 AXI チャネルは単方向で情報を転送するため、複数チャネル間の固定的な関係を必要とする要件はありません。したがって、1 サイクルの余分なレイテンシが発生するものの、どのチャネルにもレジスタスライスを挿入することができます。これにより、レイテンシのサイクル数と最大動作周波数のトレードオフが可能となります。

特定の相互接続内のほぼどの位置でもレジスタスライスを使用することが可能です。プロセッサと高性能メモリとの間には直接かつ高速の接続を使用する方が効果的ですが、パフォーマンスがそれほど重視されないペリフェラルへの長いパスは単純なレジスタスライスを使用して独立させた方が効果的です。

1.3 基本トランザクション

このセクションでは、基本的な AXI プロトコルのトランザクション例を紹介します。どの例にも **VALID** と **READY** のハンドシェイクメカニズムが使用されています。**VALID** シグナルと **READY** シグナルの両方が **HIGH** になると、アドレス情報またはデータのどちらかの転送が発生します。以下のセクションで例が説明されています。

- 読み出しバーストの例
- 読み出しバーストがオーバーラップする例: P. 1-8
- 書き込みバーストの例: P. 1-9

このセクションでは、トランザクションの順序についても 1-9 ページで説明します。

1.3.1 読み出しバーストの例

図 1-4 は、4 つの転送から成る読み出しバーストを示しています。この例では、マスタによってアドレスが生成され、スレーブが 1 サイクル後にこのアドレスを受け取ります。

注

また、バーストの長さとタイプを示す制御シグナル群もマスタによって生成されますが、図を見やすくするためにこれらのシグナルは省略されています。

アドレスバス上にアドレスが生成されたあと、読み出しデータチャンネル上でデータ転送が発生します。読み出しデータの準備ができるまで、スレーブは **VALID** シグナルを **LOW** でホールドします。バーストの最終データ転送が実行されると、スレーブは **RLAST** シグナルをアサートすることによって最後のデータ項目が転送されることを通知します。

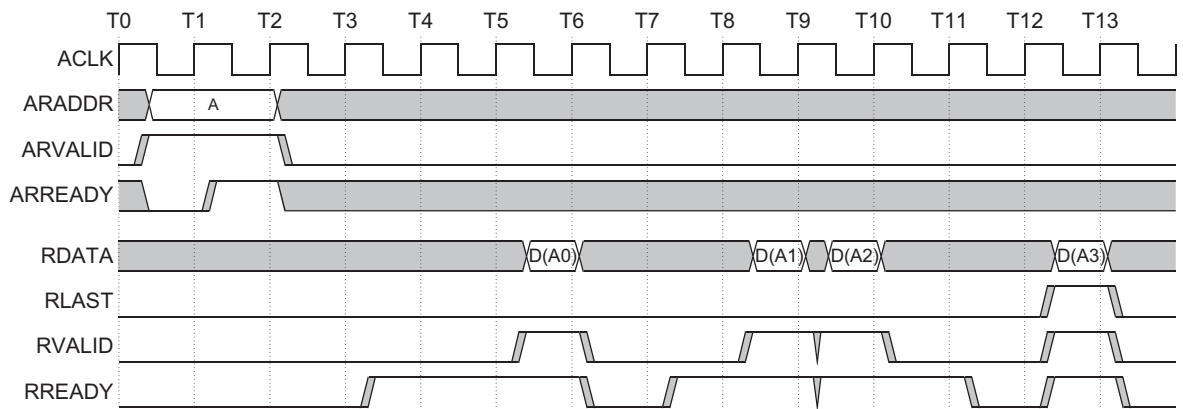


図 1-4 読み出しバースト

1.3.2 読み出しバーストがオーバーラップする例

図 1-5 は、スレーブが最初のアドレスを受け取った後で、マスタによって別のバーストが実行される様子を示しています。スレーブは、最初のバーストの完了と並行して 2 番目のバーストのデータ処理を開始できます。

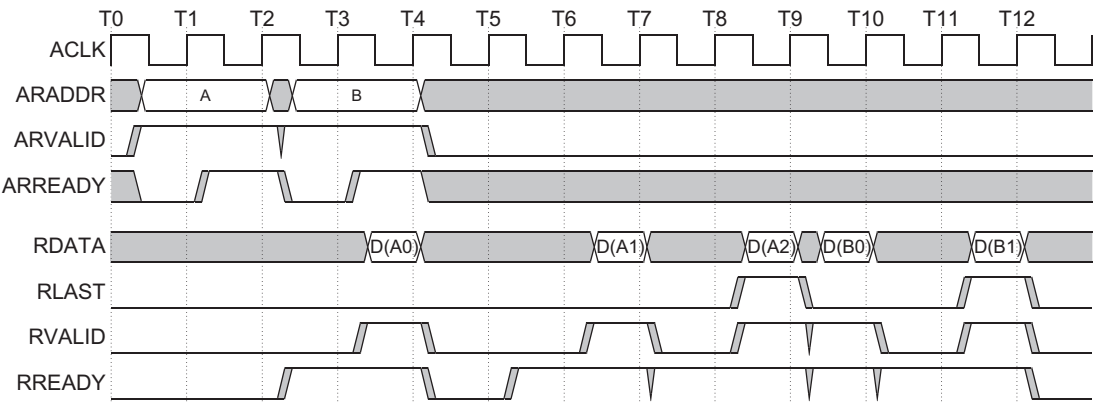


図 1-5 読み出しバーストのオーバーラップ

1.3.3 書き込みバーストの例

図 1-6 は、書き込みトランザクションを示しています。書き込みトランザクションのプロセスは、マスタがアドレス情報と制御情報を書き込みアドレスチャネルに送信した時点で開始されます。次にマスタは書き込みデータの各項目を書き込みデータチャネルを介して送信します。マスタによって最後のデータ項目が送信されるとき、**WLAST** シグナルが **HIGH** に変化します。スレーブはすべてのデータ項目を受信すると、書き込み応答シグナルをマスタに送ることによって書き込みトランザクションの完了を通知します。

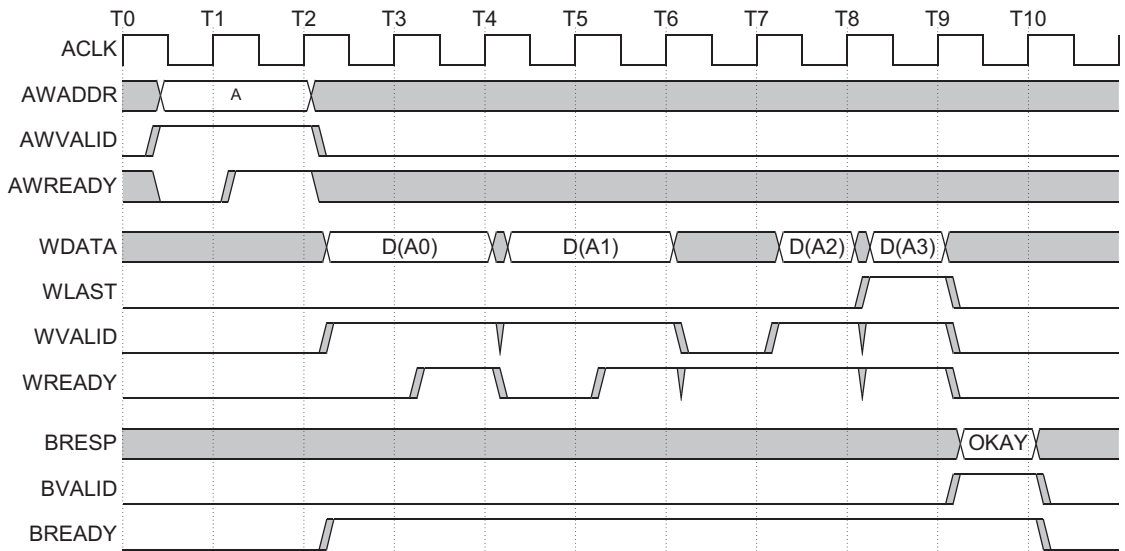


図 1-6 書き込みバースト

1.3.4 トランザクションの順序

AXI プロトコルでは、Out-of-Order でトランザクションを完了できます。AXI プロトコルでは、このインタフェースを介して実行される各トランザクションに ID タグが付けられます。AXI プロトコルでは、同一 ID タグをもつトランザクションは順番に完了する必要がありますが、異なる ID タグをもつトランザクションは Out-of-Order の実行が許されます。

Out-of-Order トランザクションによって、以下の 2 通りの方法でシステムのパフォーマンスを上げることができます。

- 相互接続により、前に実行された遅いスレーブを使用したトランザクションよりも先に、応答が速いスレーブが使用されたトランザクションを完了できます。

- 複合スレーブによって、読み出しデータをアウトオブオーダーで返すことができます。たとえば、前のアクセスのデータが取得される前に、後のアクセスのデータを内部バッファから取得することが考えられます。

すべてのトランザクションを発行された順序と同じ順序で完了することがマスタによって要求される場合には、これらのトランザクションには同じ ID タグを付ける必要があります。しかし、トランザクションを順番に完了することがマスタによって要求されない場合には、トランザクションに異なる ID タグを付けることによって、任意の順序で完了することができます。

マルチマスタシステムでは、すべてのマスタからの ID タグが一意となるように、相互接続によって ID タグに特別な情報を追加する必要があります。ID タグはマスタ番号と似ていますが、仮想マスタ番号を示す ID タグを付けるという拡張により、各マスタは同じポート内で複数の仮想マスタを実装できます。

複合デバイスはこの Out-of-Order 機能を利用できますが、単純なデバイスはこの機能を使用する必要はありません。単純なマスタは各トランザクションに同じ ID タグを付けて発行し、単純なスレーブは ID タグに関係なく各トランザクションに順番に応答することができます。

1.4 その他の機能

AXI プロトコルは以下の機能もサポートしています。

バーストタイプ

AXI は、以下の処理に適した 3 つの異なるバーストタイプをサポートしています。

- 通常のメモリアクセス
- キャッシュラインをラップするバースト
- ペリフェラルの FIFO 位置へ送られるストリーミングデータ

第 4 章「アドレッシングオプション」を参照して下さい。

システムキャッシュのサポート

AXI プロトコルのキャッシュサポートシグナルにより、マスタはシステムレベルのキャッシュに対し、トランザクションのバッファ可能属性、キャッシュ可能属性、ならびに割り当て属性を指定することができます。

P. 5-2 「キャッシュのサポート」を参照して下さい。

保護ユニットのサポート

特権アクセスとセキュアアクセスの両方を可能にするため、AXI プロトコルでは 3 つのレベルの保護ユニットがサポートされています。

P. 5-5 「保護ユニットのサポート」を参照して下さい。

アトミック処理

AXI プロトコルでは、排他アクセスとロックアクセスの両方のメカニズムが定義されています。

第 6 章「アトミックアクセス」を参照して下さい。

エラーサポート

AXI プロトコルでは、アドレスデコードエラーとスレーブ生成エラーの両方がサポートされています。

第 7 章「応答シグナル」を参照して下さい。

アンアラインドアドレス

バースト内での最初のアクセスのパフォーマンスを上げるため、AXI プロトコルでは非境界整列のバースト開始アドレスがサポートされています。

第 10 章「アンアラインド転送」を参照して下さい。

はじめに

第 2 章

シグナルの説明

本章では AXI シグナルの定義について説明します。バスの幅とトランザクション ID の幅は実装によって異なりますが、本章の表には 32 ビットのデータバス、4 ビットの書き込みデータストローブ、4 ビットの ID フィールドが使用されています。本章は以下のセクションから構成されています。

- グローバルシグナル : P. 2-2
- 書き込みアドレスチャネルシグナル : P. 2-3
- 書き込みデータチャネルシグナル : P. 2-4
- 書き込み応答チャネルシグナル : P. 2-5
- 読み出しアドレスチャネルシグナル : P. 2-6
- 読み出しデータチャネルシグナル : P. 2-7
- 低電力インタフェースシグナル : P. 2-8

2.1 グローバルシグナル

表 2-1 は、グローバル AXI シグナルを示しています。

表 2-1 グローバルシグナル

シグナル	ソース	説明
ACLK	クロックソース	グローバルクロックシグナル。すべてのシグナルは、このグローバルクロックの立ち上がりエッジでサンプリングされます。.
ARESETn	リセットソース	グローバルクロックシグナル。このシグナルは LOW でアクティブになります。

2.2 書き込みアドレスチャネルシグナル

表 2-2 は、AXI 書き込みアドレスチャネルシグナルを示しています。

表 2-2 書き込みアドレスチャネルシグナル

シグナル	ソース	説明
AWID[3:0]	マスタ	書き込みアドレス ID です。このシグナルは、書き込みアドレスシグナル群用の ID タグです。
AWADDR[31:0]	マスタ	書き込みアドレスです。この書き込みアドレスバスによって、バースト内の最初の転送アドレスが示されます。関連する制御シグナルが、バースト内の残りの転送アドレスを決定するために使用されます。
AWLEN[3:0]	マスタ	バースト長です。バースト長で、バースト内の正確な転送回数を指定します。この情報によって、指定されたアドレスに関連するデータ転送回数が決まります。P. 4-3 表 4-1 を参照して下さい。
AWSIZE[2:0]	マスタ	バーストサイズです。このシグナルは、バースト内の各転送のサイズを示します。バイトレーンストロープによって、更新される正確なバイトレーンがわかります。P. 4-4 表 4-2 を参照して下さい。
AWBURST[1:0]	マスタ	バーストタイプです。バーストタイプはサイズ情報と組み合わせて使用され、バースト内の各転送のサイズを計算する方法を指定します。P. 4-5 表 4-3 を参照して下さい。
AWLOCK[1:0]	マスタ	ロックタイプです。このシグナルは、転送のアトミックな性質に関する情報を提供します。P. 6-2 表 6-1 を参照して下さい。
AWCACHE[3:0]	マスタ	キャッシュタイプです。このシグナルは、トランザクションの属性として「パッファ可能」、「キャッシュ可能」、「ライトスルー」、「ライトバック」、「割り当て」のいずれかを示します。P. 5-3 表 5-1 を参照して下さい。
AWPROT[2:0]	マスタ	保護タイプです。このシグナルは、トランザクションの保護レベルが「通常」、「特権」、「セキュア」のいずれであるかと、そのトランザクションがデータアクセスと命令アクセスのどちらであることを示します。P. 5-5 「保護ユニットのサポート」を参照して下さい。
AWVALID	マスタ	書き込みアドレスが有効であることを示します。このシグナルは、書き込みアドレス情報と制御情報が有効であることを示します。 1 = アドレス情報と制御情報が有効 0 = アドレス情報と制御情報が無効 アドレス情報と制御情報は、アドレス応答シグナル AWREADY が HIGH に変わるまで変化しません。
AWREADY	スレーブ	書き込みアドレスのための準備が整ったことを示します。このシグナルは、スレーブがアドレスとその制御シグナルを受信できる状態にあるかどうかを示します。 1 = スレーブ準備完了 0 = スレーブ準備中

2.3 書き込みデータチャネルシグナル

表 2-3 は、AXI 書き込みデータチャネルシグナルを示しています。

表 2-3 書き込みデータチャネルシグナル

シグナル	ソース	説明
WID[3:0]	マスタ	書き込み ID タグです。このシグナルは書き込みデータ転送の ID タグを示します。 WID の値は、書き込みトランザクションの AWID の値と一致している必要があります。
WDATA[31:0]	マスタ	書き込みデータです。書き込みデータバスの幅には 8、16、32、64、128、256、512、1024 ビットのいずれかを指定できます。
WSTRB[3:0]	マスタ	書き込みストローブです。このシグナルは、メモリ内で更新するバイトレーンを示します。書き込みデータバスの 8 ビットごとに 1 つの書き込みストローブがあります。したがって、 WSTRB[n] は WDATA[(8×n)+7:(8×n)] に対応します。
WLAST	マスタ	最後の書き込みを示します。このシグナルは、書き込みバースト内の最後の転送であることを通知します。
WVALID	マスタ	書き込みが有効であることを示します。このシグナルは、有効な書き込みデータと書き込みストローブが存在することを示します。 1 = 書き込みデータと書き込みストローブが存在します。 0 = 書き込みデータと書き込みストローブが存在しません。
WREADY	スレーブ	書き込みのための準備が整ったことを示します。このシグナルは、スレーブが書き込みデータを受信できる状態にあるかどうかを通知します。 1 = スレーブ準備完了 0 = スレーブ準備中

2.4 書き込み応答チャネルシグナル

表 2-4 は、AXI 書き込み応答チャネルシグナルを示しています。

表 2-4 書き込み応答チャネルシグナル

シグナル	ソース	説明
BID[3:0]	スレーブ	応答 ID を示します。書き込み応答の識別タグです。BID の値は、そのスレーブが 応答する書き込みトランザクションのAWIDの値と一致している必要があります。
BRESP[1:0]	スレーブ	書き込み応答です。このシグナルは書き込みトランザクションのステータスを示 します。応答にはOKAY、EXOKAY、SLVERR、DECERR の 4 つがあります。
BVALID	スレーブ	書き込み応答が有効であることを示します。このシグナルは、有効な書き込み応 答が存在することを示します。 1 = 書き込み応答あり 0 = 書き込み応答なし
BREADY	マスタ	応答の準備が整っていることを示します。このシグナルは、マスタが応答情報を 受信できる状態にあるかどうかを通知します。 1 = マスタ準備完了 0 = マスタ準備

2.5 読み出しアドレスチャネルシグナル

P. 2-3 表 2-2 は、AXI 読み出しアドレスチャネルシグナルを示しています。

表 2-5 読み出しアドレスチャネルシグナル

シグナル	ソース	説明
ARID[3:0]	マスタ	読み出しアドレス ID です。このシグナルは、読み出しアドレスシグナル群用の ID タグです。
ARADDR[31:0]	マスタ	読み出しアドレスです。この読み出しアドレスバスによって、バースト内の最初の転送アドレスが示されます。バーストの開始アドレスだけが通知されますが、アドレスとともに発行される制御シグナルによって、バースト内の残りの転送の、アドレスの計算方法が指定されます。
ARLEN[3:0]	マスタ	バースト長です。バースト長で、バースト内の正確な転送回数を指定します。この情報によって、指定されたアドレスに関連するデータ転送回数が決まります。P. 4-3 表 4-1 を参照して下さい。
ARSIZE[2:0]	マスタ	バーストサイズです。このシグナルは、バースト内の各転送のサイズを示します。P. 4-4 表 4-2 を参照して下さい。
ARBURST[1:0]	マスタ	バーストタイプです。バーストタイプはサイズ情報と組み合わせて使用され、バースト内の各転送のサイズを計算する方法を指定します。P. 4-5 表 4-3 を参照して下さい。
ARLOCK[1:0]	マスタ	ロックタイプです。このシグナルは、転送のアトミックな性質に関する情報を提供します。P. 6-2 表 6-1 を参照して下さい。
ARCACHE[3:0]	マスタ	キャッシュタイプです。このシグナルは、転送のキャッシュ特性に関する情報を提供します。P. 5-3 表 5-1 を参照して下さい。
ARPROT[2:0]	マスタ	保護タイプです。このシグナルは、トランザクションに関する保護ユニットの情報を提供します。P. 5-5 「保護ユニットのサポート」を参照して下さい。
ARVALID	マスタ	読み出しアドレスが有効であることを示します。このシグナルが HIGH の場合には、読み出しアドレス情報と制御情報が有効であり、アドレス応答シグナル ARREADY が HIGH になるまで変化しないことを示します。 1 = アドレス情報と制御情報が有効 0 = アドレス情報と制御情報が無効
ARREADY	スレーブ	読み出しアドレスのための準備が整ったことを示します。このシグナルは、スレーブがアドレスとその制御シグナルを受信できる状態にあるかどうかを示します。 1 = スレーブ準備完了 0 = スレーブ準備中

2.6 読み出しデータチャネルシグナル

表 2-6 は、AXI 読み出しデータチャネルシグナルを示しています。

表 2-6 読み出しデータチャネルシグナル

シグナル	ソース	説明
RID[3:0]	スレーブ	読み出し ID タグです。このシグナルは、読み出しデータシグナル群の ID タグです。 RID の値はスレーブによって生成されます。この値は、そのスレーブが応答する読み出しトランザクションの ARID の値と一致している必要があります。
RDATA[31:0]	スレーブ	読み出しデータです。読み出しデータバスの幅には 8、16、32、64、128、256、512、1024 ビットのいずれかを指定できます。
RRESP[1:0]	スレーブ	読み出し応答です。このシグナルは読み出し転送のステータスを通知します。応答には OKAY 、 EXOKAY 、 SLVERR 、 DECERR の 4 つがあります。
RLAST	スレーブ	最後の読み出しを示します。このシグナルは、読み出しバースト内の最後の転送であることを示します。
RVALID	スレーブ	読み出しが有効であることを示します。このシグナルは、要求された読み出しデータが有効で、読み出し転送を完了させることが可能なことを示しています。 1 = 読み出しデータが有効 0 = 読み出しデータが無効
RREADY	マスタ	読み出しの準備が整っていることを示します。このシグナルは、マスタが読み出しデータと応答情報を受信できる状態にあることを通知します。 1 = マスタ準備完了 0 = マスタ準備中

2.7 低電力インタフェースシグナル

表 2-7 は、オプションの低電力インタフェースのシグナルを示しています。

表 2-7 低電力インタフェースシグナル

シグナル	ソース	説明
CSYSREQ	クロック コントローラ	システムが低電力で動作する必要があることを示します。システムのクロックコントローラはこのシグナルを使用して、ペリフェラルに低電力状態に入るよう要求します。
CSYSACK	ペリフェラル デバイス	低電力要求の確認応答を示します。このシグナルは、システムの低電力要求に対するペリフェラルからの確認応答です。
CACTIVE	ペリフェラル デバイス	クロックがアクティブであることを示します。このシグナルは、ペリフェラルがクロッキングナルを必要としてることを示します。 1 = ペリフェラルのクロックが必要 0 = ペリフェラルのクロックは不用

第 3 章

チャンネルハンドシェイク

本章では、マスタ / スレーブのハンドシェイクプロセスと、**READY** および **VALID** ハンドシェイクシグナルの関係とデフォルト値について説明します。本章は以下のセクションから構成されています。

- ハンドシェイクプロセス : P. 3-2
- チャンネル間の関係 : P. 3-6
- チャンネルハンドシェイクシグナル間の依存関係 : P. 3-7

3.1 ハンドシェイクプロセス

5つのチャンネルはすべて同じ **VALID/READY** ハンドシェイクシグナルを使用してデータと制御情報を転送します。マスタとスレーブはこの 2 ウェイフロー制御メカニズムによって、データと制御情報の転送速度を制御することができます。ソースは **VALID** シグナルを生成して、有効なデータまたは制御情報が有効になったタイミングを通知します。デスティネーションは **READY** シグナルを生成して、データまたは制御情報を受信可能な状態であることを通知します。**VALID** シグナルと **READY** シグナルの両方が **HIGH** にセットされた場合にのみ転送が発生します。

マスタとスレーブの両インタフェース上の入力シグナルと出力シグナルの間には組み合わせパスがあつてはなりません。

図 3-1 から P. 3-3 図 3-3 は、ハンドシェイクシーケンスの例を示しています。図 3-1 では、ソースがデータまたは制御情報を供給し、**VALID** シグナルを **HIGH** でドライブしています。ソースからのデータまたは制御情報は、デスティネーションが **READY** シグナルを **HIGH** でドライブし、データまたは制御情報を受信可能であることを通知するまで、状態が保持されます。矢印は転送発生時を示しています。

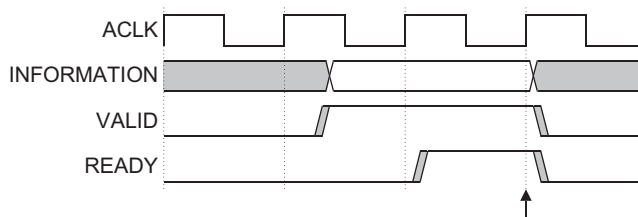


図 3-1 **VALID** の後に **READY** が通知されるハンドシェイク

P. 3-3 図 3-2 では、データあるいは制御情報が有効になる前に、デスティネーションが **READY** シグナルを **HIGH** でドライブしています。このシグナルは、データまたは制御情報が有効になると同時に、デスティネーションがそのデータまたは制御情報を 1 サイクルで受信できることを示します。矢印は転送発生時を示しています。

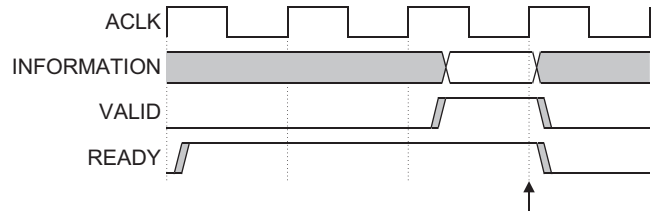


図 3-2 READY の後に VALID が通知されるハンドシェイク

図 3-3 は、ソースとデスティネーションの両方が偶然に同じサイクルでデータまたは制御情報の転送が可能であることを通知するケースを示しています。この場合、転送は即時に発生します。矢印は転送発生時を示しています。

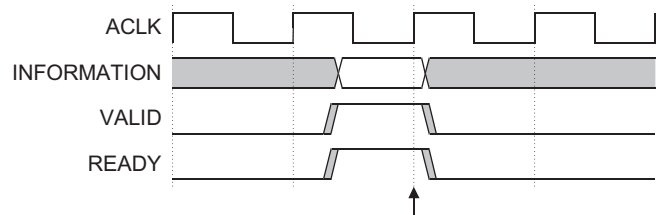


図 3-3 VALID と READY が同時に通知されるハンドシェイク

AXI プロトコルチャンネルハンドシェイクの個々の機構は以下のセクションで説明されています。

- 書き込みアドレスチャンネル
- 書き込みデータチャンネル：P. 3-4
- 書き込み応答チャンネル：P. 3-4
- 読み出しアドレスチャンネル：P. 3-4
- 読み出しデータチャンネル：P. 3-5

3.1.1 書き込みアドレスチャンネル

マスタは、有効なアドレスおよび制御情報を送出する場合にのみ **AWVALID** シグナルをアサートできます。**AWVALID** シグナルは、スレーブがアドレス情報と制御情報を受信して **AWREADY** シグナルをアサートするまでアサートされ続ける必要があります。

AWREADY のデフォルト値には HIGH または LOW のどちらでもセットできます。推奨されるデフォルト値は HIGH ですが、**AWREADY** が HIGH の場合には、スレーブは渡される有効なアドレスは何でも受信できなくてはなりません。

AWREADY のデフォルト値に **LOW** をセットすることも可能ですが、この場合は **AWVALID** と **AWREADY** のアサートにそれぞれ 1 サイクルが必要となり、転送に少なくとも 2 サイクルが必要とされるため、この方法は推奨されません。

3.1.2 書き込みデータチャンネル

書き込みバーストの実行時、マスタは有効な書き込みデータを送出する場合にのみ **WVALID** シグナルをアサートできます。**WVALID** は、スレーブがその書き込みデータを受信して **WREADY** シグナルをアサートするまでアサートされ続ける必要があります。

WREADY のデフォルト値は、スレーブが書き込みデータを常に 1 サイクルで受信できる場合にのみ **HIGH** にセットできます。

マスタは、バースト内の最後の書き込み転送をドライブするときに **WLAST** シグナルをアサートする必要があります。

WVALID が **LOW** の場合、**WSTRB[3:0]** シグナルは任意の値を取ることができますが、これらのシグナルは **LOW** でドライブされるか、前の値でホールドされる設定が推奨です。

3.1.3 書き込み応答チャンネル

スレーブは、有効な書き込み応答を生成する場合にのみ **BVALID** シグナルをアサートできます。**BVALID** は、マスタがその書き込み応答を受信して **BREADY** シグナルをアサートするまでアサートされ続ける必要があります。

BREADY のデフォルト値は、マスタが書き込み応答を常に 1 サイクルで受信できる場合にのみ **HIGH** にセットできます。

3.1.4 読み出しアドレスチャンネル

マスタは、有効なアドレスおよび制御情報を送出する場合にのみ **ARVALID** シグナルをアサートできます。**ARVALID** シグナルは、スレーブがアドレス情報と制御情報を受信して **ARREADY** シグナルをアサートするまでアサートされ続ける必要があります。

ARREADY のデフォルト値には **HIGH** または **LOW** のどちらでもセットできます。推奨されるデフォルト値は **HIGH** ですが、**ARREADY** が **HIGH** の場合には、スレーブは渡される有効なアドレスのすべてを受信できなくてはなりません。

ARREADY のデフォルト値に **LOW** をセットすることも可能ですが、この場合は **ARVALID** と **ARREADY** のアサートにそれぞれ 1 サイクルが必要となり、転送に少なくとも 2 サイクルが必要とされるため、この方法は推奨されません。

3.1.5 読み出しデータチャンネル

スレーブは、有効な読み出しデータを送出する場合にのみ **RVALID** シグナルをアサートできます。**RVALID** は、マスタがその読み出しデータを受信して **RREADY** シグナルをアサートするまでアサートされ続ける必要があります。スレーブに読み出しデータのソースが1つしかない場合でも、スレーブはそのデータの要求に応答して **RVALID** シグナルをアサートする必要があります。

マスタは**RREADY**シグナルを使用して、データを受信することを通知します。**RREADY** のデフォルト値は、読み出しトランザクションの実行時にマスタが常に読み出しデータをすぐに受信できる場合にのみ **HIGH** にセットできます。

スレーブは、バースト内の最後の読み出し転送を実行するときに **RLAST** シグナルをアサートする必要があります。

3.2 チャンネル間の関係

「アドレス」、「読み出し」、「書き込み」、「書き込み応答」のチャンネル間の関係は一定ではありません。

たとえば書き込みデータは、その書き込みデータのアドレスの前にインタフェース上に出現することがあります。このケースは、書き込みアドレスチャンネルに書き込みデータチャンネルよりも多くのレジスタステージがある場合に発生し得ます。また、書き込みデータがアドレスと同じサイクル内に出現する場合もあります。

相互接続によってデスティネーションのアドレス空間またはスレーブ空間が決定される必要がある場合、そのアドレスと書き込みデータは相互接続によって整頓される必要があります。書き込みデータが転送先に指定されているスレーブに対してのみ有効と通知するためにはこれが必要となります。

以下の2つの関係は必ず維持する必要があります。

- 読み出しデータは、必ずそのデータに関連するアドレスの後に生成される必要があります。
- 書き込み応答は、その書き込み応答に関連する書き込みトランザクション内の最後の書き込み転送に続いて生成される必要があります。

3.3 チャンネルハンドシェイクシグナル間の依存関係

デッドロック状態を回避するには、ハンドシェイクシグナル間の依存関係に注意する必要があります。

すべてのトランザクションにおいて以下のような状態である必要があります。

- 1つのAXIコンポーネントの**VALID**シグナルは、そのトランザクションにおけるもう一方のコンポーネントの**READY**シグナルに依存してはなりません。
- **READY**シグナルは、**VALID**シグナルがアサートされるまで待機できます。

注

READYシグナルがアサートされる前に、**VALID**シグナルのアサートを待つことができるほか、**VALID**シグナルのアサートの前にデフォルトで**READY**シグナルをアサートすることもできます。そのため、効率的な設計が可能です。

図3-4とP.3-8図3-5は、ハンドシェイクシーケンスの依存関係を示しています。単頭矢印は、前のシグナルがアサートされる前または後にアサートできるシグナルを指しています。二頭矢印は、前のシグナルがアサートされた後にしかアサートできないシグナルを指しています。

図3-4は、読み出しトランザクションにおけるシグナルの依存関係を示しています。

- スレーブは、**ARREADY**をアサートする前に、**ARVALID**がアサートされるのを待つことができます。
- スレーブは、**RVALID**をアサートして読み出しデータを返し始める前に、**ARVALID**と**ARREADY**の両方がアサートされるのを待つ必要があります。

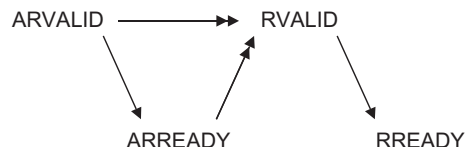


図3-4 読み出しトランザクションにおけるハンドシェイクの依存関係

P.3-8図3-5は、書き込みトランザクションにおけるシグナルの依存関係を示しています。

- マスタは、スレーブによる**AWREADY**あるいは**WREADY**のアサートを待たずに**AWVALID**あるいは**WVALID**をアサートする必要があります。
- スレーブは、**AWREADY**をアサートする前に、**AWVALID**と**WVALID**のいずれかあるいは両方がアサートされるのを待つことができます。

- スレーブは、**WREADY** をアサートする前に、**AWVALID** と **WVALID** のいずれかあるいは両方がアサートされるのを待つことができます。
- スレーブは、**BVALID** をアサートする前に、**WVALID** と **WREADY** の両方がアサートされるのを待つ必要があります。

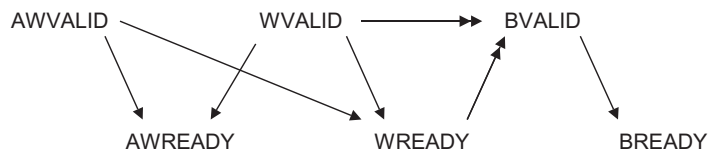


図 3-5 書き込みトランザクションにおけるハンドシェイクの依存関係

注

書き込みトランザクションの間、マスタが **WVALID** を駆動するまえに **AWREADY** がアサートされるのを待つてはいけないということは重要です。もしスレーブが逆に **AWREADY** をアサートするまえに **WVALID** を待っているとデッドロックを引き起こす可能性があるからです。

第 4 章

アドレッシングオプション

本章では、AXI のバーストタイプと、バースト転送に使用されるアドレスおよびバイトレーンの計算方法について説明します。本章は以下のセクションから構成されています。

- アドレッシングオプションについて : P. 4-2
- バースト長 : P. 4-3
- バーストサイズ : P. 4-4
- バーストタイプ : P. 4-5
- バーストアドレス : P. 4-7

4.1 アドレッシングオプションについて

AXI はバースト指向のプロトコルです。マスタは転送制御情報とその転送の先頭バイトのアドレスを指定することによってバーストを開始します。バーストトランザクションが実行される間、バースト内のその後の転送アドレスはスレーブによって計算される必要があります。

バーストがスレーブ間の境界を超えないようにし、スレーブ内で必要とされるアドレスインクリメンタのサイズを抑えるために、バーストが 4KB 境界を超えないようにする必要があります。

4.2 バースト長

AWLEN シグナルまたは **ARLEN** シグナルによって、各バースト内で発生するデータ転送の回数が指定されます。表 4-1 が示すように、各バーストでは 1 ～ 16 回の転送が行われます。

表 4-1 バースト長のエンコーディング

ARLEN[3:0] AWLEN[3:0]	データ 転送回数
b0000	1
b0001	2
b0010	3
⋮	
⋮	
⋮	
b1101	14
b1110	15
b1111	16

ラッピングバーストの場合、バーストの長さには 2、4、8、16 転送のいずれかを指定する必要があります。

各トランザクションは、**ARLEN** または **AWLEN** で指定される転送数で構成される必要があります。バーストを早期終了してデータ転送の数を減らすことのできるコンポーネントはありません。書き込みバースト時、マスタはすべての書き込みストロープのアサートを解除することによってその後の書き込みをディセーブルできますが、バースト内の残りの転送は完了する必要があります。読み出しバースト時、マスタはその後の読み出しデータを破棄することができますが、バースト内の残りの転送は完了する必要があります。

注意

不要な読み出しデータを破棄すると、FIFO などの読み出しが動作に影響を与えるデバイスへのアクセス時にデータが失われる可能性があります。マスタはこのようなデバイスにアクセスする場合に、必要以上のバースト長を使用してはなりません。

4.3 バーストサイズ

表 4-2 は、**ARSIZE** シグナルまたは **AWSIZE** シグナルによって、バーストにおける各データ転送で転送される最大データのバイト数が指定される方法を示しています。

表 4-2 バーストサイズのエンコーディング

ARSIZE[2:0] AWSIZE[2:0]	転送における バイト数
b000	1
b001	2
b010	4
b011	8
b100	16
b101	32
b110	64
b111	128

AXI は、各転送に使用するデータバスのバイトレーンを転送アドレスから決定します。

転送サイズがデータバスよりも小さいインクリメントバーストまたはラッピングバーストの場合、データ転送はそのバーストの単位ごとに異なるバイトレーン上で実行されます。固定バーストのアドレスは一定であり、各転送に同じバイトレーンが使用されます。

どの転送のサイズも、トランザクションの対象となるコンポーネントのデータバス幅を超えてはなりません。

4.4 バーストタイプ

AXI プロトコルでは、以下の 3 つのバーストタイプが定義されています。

- 固定バースト
- インクリメントバースト
- ラッピングバースト : P. 4-6

表 4-3 は、**ARBURST** シグナルと **AWBURST** シグナルによるバーストタイプの選択方法を示しています。

表 4-3 バーストタイプのエンコーディング

ARBURST[1:0] AWBURST[1:0]	バーストタイプ 説明		アクセス対象
b00	固定	固定アドレスバースト	FIFO 型のメモリ
b01	インクリメント	インクリメントアドレスバースト	通常のシーケンシャルなメモリ
b10	ラッピング	ラップ境界で下位アドレスにラップするアドレスインクリメントバースト	キャッシュライン
b11	予約	-	-

4.4.1 固定バースト

固定バーストでは、バースト内の各転送に同じアドレスが使用されます。このバーストタイプは、ペリフェラルの FIFO の読み出し / 書き込みなど、同じ位置に繰り返しアクセスする場合に使用されます。

4.4.2 インクリメントバースト

インクリメントバーストでは、バースト内の各転送で、前の転送アドレスに値が加算されたアドレスが使用されます。加算される値は、転送のサイズによって異なります。たとえば、サイズが 4 バイトのバーストにおける各転送アドレスには、前のアドレス +4 の値が使用されます。

4.4.3 ラッピングバースト

ラッピングバーストは、バースト内の各転送に、前のアドレスに値が加算されたアドレスが使用されるインクリメントバーストと似ています。しかしラッピングバーストでは、ラップ境界に到達すると、アドレスが下位アドレスにラップされます。ラップ境界は、バースト内の各転送のサイズに、そのバースト内の全転送数を乗算した値となります。

ラッピングバーストには以下の2つの制約があります。

- 開始アドレスは転送サイズの境界で整列される必要があります。
- バーストの長さは、2、4、8、16のいずれかである必要があります。

4.5 バーストアドレス

このセクションでは、バースト内の転送に使用されるアドレスとバイトレーンを計算する簡単な式を紹介します。これらの式には以下の変数が使用されます。

Start_Address	マスタによって発行される開始アドレスです。
Number_Bytes	各データ転送の最大バイト数です。
Data_Bus_Bytes	データバス内のバイトレーンの数です。
Aligned_Address	境界整列された開始アドレスです。
Burst_Length	バースト内のデータ転送の総数です。
Address_N	バースト内の転送Nのアドレスです。Nは2から16の整数値です。
Wrap_Boundary	ラッピングバースト内の最下位アドレスです。
Lower_Byte_Lane	転送の最下位アドレスにあるバイトのバイトレーンです。
Upper_Byte_Lane	転送の最上位アドレスにあるバイトのバイトレーンです。
INT(x)	xの切り捨て後の整数値です。

以下の式を使用して、バースト内の転送アドレスを計算することができます。

- $\text{Start_Address} = \text{ADDR}$
- $\text{Number_Bytes} = 2^{\text{SIZE}}$
- $\text{Burst_Length} = \text{LEN} + 1$
- $\text{Aligned_Address} = (\text{INT}(\text{Start_Address} / \text{Number_Bytes})) \times \text{Number_Bytes}$.

以下の式を使用して、バースト内の最初の転送アドレスを計算することができます。

- $\text{Address_1} = \text{Start_Address}$.

以下の式を使用して、バースト内の2番目以降の転送アドレスを計算することができます。

- $\text{Address_N} = \text{Aligned_Address} + (\text{N} - 1) \times \text{Number_Bytes}$.

ラッピングバーストの場合、ラップ境界を計算に入れるために Wrap_Boundary 変数が拡張されます。

- $\text{Wrap_Boundary} = (\text{INT}(\text{Start_Address} / (\text{Number_Bytes} \times \text{Burst_Length}))) \times (\text{Number_Bytes} \times \text{Burst_Length})$.

$\text{Address_N} = \text{Wrap_Boundary} + (\text{Number_Bytes} \times \text{Burst_Length})$ の場合には、以下の式を使用します。

- $\text{Address_N} = \text{Wrap_Boundary}$.

以下の式を使用して、バースト内の最初の転送に使用されるバイトレーンを計算することができます。

- $\text{Lower_Byte_Lane} = \text{Start_Address} - (\text{INT}(\text{Start_Address} / \text{Data_Bus_Bytes})) \times \text{Data_Bus_Bytes}$
- $\text{Upper_Byte_Lane} = \text{Aligned_Address} + (\text{Number_Bytes} - 1) - (\text{INT}(\text{Start_Address} / \text{Data_Bus_Bytes})) \times \text{Data_Bus_Bytes}$.

以下の式を使用して、バースト内の 2 番目以降の転送に使用されるバイトレーンを計算することができます。

- $\text{Lower_Byte_Lane} = \text{Address_N} - (\text{INT}(\text{Address_N} / \text{Data_Bus_Bytes})) \times \text{Data_Bus_Bytes}$
- $\text{Upper_Byte_Lane} = \text{Lower_Byte_Lane} + \text{Number_Bytes} - 1$.

データは以下のバイトレーンで転送されます。

- $\text{DATA}[(8 \times \text{Upper_Byte_Lane}) + 7 : (8 \times \text{Lower_Byte_Lane})]$.

第 5 章

その他の制御情報

本章では、システムレベルのキャッシュと保護ユニットに関する AXI プロトコルのサポートについて説明します。本章は以下のセクションから構成されています。

- キャッシュのサポート : P. 5-2
- 保護ユニットのサポート : P. 5-5

5.1 キャッシュのサポート

システムレベルのキャッシュと、性能を向上させるためのコンポーネントをサポートするために、**ARCACHE** と **AWCACHE** のキャッシュ情報シグナルが用意されています。これらのシグナルによって、トランザクションの処理方法に関する情報が得られます。

ARCACHE[3:0] シグナルまたは **AWCACHE[3:0]** シグナルによって、トランザクションに「バッファ可能」、「キャッシュ可能」、「割り当て」のいずれかの属性が指定されます。これにより、システムレベルのキャッシュがサポートされます。

バッファ可能 (B) ビット : ARCACHE[0] および AWCACHE[0]

バッファ可能ビットが **HIGH** にセットされている場合、相互接続またはどのコンポーネントでも、その最終転送先に到達しようとしているトランザクションを任意のサイクル数の間、遅延させることができます。これは通常、書き込みの場合に限られます。

キャッシュ可能 (C) ビット : ARCACHE[1] および AWCACHE[1]

キャッシュ可能ビットが **HIGH** にセットされている場合、デスティネーションでのトランザクションの特性が、元のトランザクションの特性と一致している必要はありません。

書き込みトランザクションの場合は、複数の書き込みを結合することができます。

読み出しトランザクションの場合には、読み出し位置のプリフェッチを行うことができるか、あるいは複数の読み出しトランザクションに対して1度だけのフェッチですますことができます。

トランザクションがキャッシュされるかどうか判断するには、このビットを、読み出し割り当て (RA) ビットと書き込み割り当て (WA) ビットとともに使用する必要があります。

読み出し割り当て (RA) ビット : ARCACHE[2] および AWCACHE[2]

読み出し割り当て (RA) ビットが **HIGH** にセットされている場合は、読み出しアクセスでキャッシュミスが発生すると、そのトランザクションがキャッシュに割り当てられることを示しています。

キャッシュ可能ビットが **LOW** の場合は、読み出し割り当てビットを **HIGH** にセットすることはできません。

書き込み割り当て (WA) ビット : ARCACHE[3] および AWCACHE[3]

書き込み割り当て (WA) ビットが **HIGH** にセットされている場合は、書き込みアクセスでキャッシュミスが発生すると、そのトランザクションがキャッシュに割り当てられることを示しています。

キャッシュ可能ビットが **LOW** の場合は、書き込み割り当てビットを **HIGH** にセットすることはできません。

表 5-1 は、**ARCCACHE[3:0]** シグナルと **AWCCACHE[3:0]** シグナルのエンコーディングを示しています。

表 5-1 キャッシュのエンコーディング

ARCCACHE[3:0] AWCCACHE[3:0]				
WA	RA	C	B	トランザクションの属性
0	0	0	0	キャッシュ不可能およびバッファリング不可能
0	0	0	1	バッファリング可能のみ
0	0	1	0	キャッシュ可能、割り当て不可
0	0	1	1	キャッシュ可能およびバッファリング可能、割り当て不可
0	1	0	0	予約
0	1	0	1	予約
0	1	1	0	キャッシュ可能ライトスルー、読み出し時のみ割り当て
0	1	1	1	キャッシュ可能ライトバック、読み出し時のみ割り当て
1	0	0	0	予約
1	0	0	1	予約
1	0	1	0	キャッシュ可能ライトスルー、書き込み時のみ割り当て
1	0	1	1	キャッシュ可能ライトバック、書き込み時のみ割り当て
1	1	0	0	予約
1	1	0	1	予約
1	1	1	0	キャッシュ可能ライトスルー、読み出し時および書き込み時に割り当て
1	1	1	1	キャッシュ可能ライトバック、読み出し時および書き込み時に割り当て

書き込みトランザクションでは、**AWCCACHE** シグナルによって書き込み応答を発行するコンポーネントが決まります。書き込みトランザクションがバッファリング可能の場合には、ブリッジやシステムレベルのキャッシュで書き込み応答を返すことができます。トランザクションがバッファ不可能の場合には、書き込み応答はトランザクションの最終デスティネーションから返す必要があります。

AXI では、バッファまたはキャッシュに格納されるデータをそのデスティネーションに到達させるメカニズムを定義していません。たとえば、システムレベルのキャッシュには、キャッシュエントリのクリーニング、フラッシュ、無効化を制御するコントローラが内蔵されている場合があります。別の例としては、一致するトランザクション ID をもつバッファ不可能な書き込みを受信したときに、バッファの内容を書き出す制御ロジックが組み込まれた、書き込みバッファを備えるブリッジが挙げられます。

5.2 保護ユニットのサポート

通常、複雑なシステム設計をサポートするには、システム内の内部接続と他のデバイスの両方に、不正トランザクションからシステムを保護する機能が必要となります。**AWPROT** シグナルまたは **ARPROT** シグナルを使用することによって 3 つのレベルのアクセス保護を実現できます。

通常または特権 : **ARPROT[0]** および **AWPROT[0]**

- LOW の場合は通常のアクセスを示します。
- HIGH の場合は特権アクセスを示します。

マスタが処理モードを示す際に使用されます。特権処理モードは通常、システムにおいて高いアクセス権を持ちます。

セキュアまたは非セキュア : **ARPROT[1]** および **AWPROT[1]**

- LOW の場合はセキュアなアクセスを示します。
- HIGH の場合はセキュアでないアクセスを示します。

処理モード間でレベルを大きく変えたいようなときに使用します。

——— 注 ———

このビットが HIGH の場合、トランザクションは非セキュアとなり、LOW の場合、トランザクションはセキュアになります。

命令またはデータ : **ARPROT[2]** および **AWPROT[2]**

- LOW の場合はデータアクセスを示します。
- HIGH の場合は命令アクセスを示します。

このビットは、トランザクションが命令アクセスかデータアクセスかを示します。

——— 注 ———

この情報はヒントとして提供されます。常に正確とは限りません。たとえば、1 つのトランザクションに命令アクセスとデータアクセスが混在する場合があります。デフォルトでは、特に命令アクセスと指定されていない限り、アクセスはデータアクセスと認識したほうがよいでしょう。

表 5-2 は、**ARPROT[2:0]** シグナルと **AWPROT[2:0]** シグナルのエンコーディングを示しています。

表 5-2 保護レベルのエンコーディング

ARPROT[2:0] AWPROT[2:0]	保護レベル
[0]	1 = 特権アクセス 0 = 通常アクセス
[1]	1 = 非セキュアアクセス 0 = セキュアアクセス
[2]	1 = 命令アクセス 0 = データアクセス

第 6 章

アトミックアクセス

本章では、排他アクセスおよびロックアクセスのメカニズムが AXI プロトコルにどのように実装されるかについて説明します。本章は以下のセクションから構成されています。

- アトミックアクセスについて : P. 6-2
- 排他アクセス : P. 6-3
- ロックアクセス : P. 6-7

6.1 アトミックアクセスについて

アトミックアクセスのプリミティブの実装のために、**ARLOCK[1:0]** シグナルと **AWLOCK[1:0]** シグナルによって排他アクセスとロックアクセスの機構が用意されています。表 6-1 は、**ARLOCK[1:0]** シグナルと **AWLOCK[1:0]** シグナルのエンコーディングを示しています。

表 6-1 アトミックアクセスのエンコーディング

ARLOCK[1:0] AWLOCK[1:0]	アクセスタイプ
b00	通常アクセス
b01	排他アクセス
b10	ロックアクセス
b11	予約

6.2 排他アクセス

排他アクセスのメカニズムが提供されているため、処理が実行される間バスを特定のマスタにロックするという方法を用いなくてもセマフォ型の処理を実装できます。排他アクセスの利点は、セマフォ型の処理が実行されても、バスアクセスレイテンシが大きくならず、帯域幅を損なうこともない点です。

排他アクセスは **ARLOCK[1:0]** シグナルか **AWLOCK[1:0]** シグナルによって選択され、その排他アクセスの成功・失敗は **RRESP[1:0]** シグナルと **BRESP[1:0]** シグナルによって通知されます (P. 7-2 表 7-1 参照)。

スレーブには、排他アクセスをサポートする特別なロジックが必要となります。AXI プロトコルには、排他アクセスをサポートしていないスレーブに対してマスタから排他アクセスが試行されたことを通知するフェールセーフメカニズムが組み込まれています。

6.2.1 排他アクセスの処理

排他アクセスの基本的な処理の流れは以下の通りです。

1. マスタによって特定のアドレス位置からの排他読み出しが実行されます。
2. その後、マスタは同じアドレス位置への排他書き込みを実行することにより、この排他アクセスの完了を試みます。
3. マスタからの排他書き込みアクセスは、以下のように通知されます。
 - 読み出しアクセスと書き込みアクセスの間に、そのアドレス位置への書き込みを行った他のマスタが存在しない場合には、「成功」が通知されます。
 - 読み出しアクセスと書き込みアクセスの間に、別のマスタがそのアドレス位置への書き込みを行った場合には、「失敗」が通知されます。この場合にはアドレス位置は更新されません。

注

マスタは排他処理の書き込み部分が完了されない場合があります。排他アクセス監視ハードウェアは、通常 1 つのトランザクション ID につき 1 つのアドレスしか監視しません。したがって、マスタが排他処理の書き込み部分を完了しなかった場合、以後の排他読み出しによって、排他性が監視されるアドレスが変更されます。

6.2.2 マスタの観点から見た排他アクセス

マスタは排他読み出しを実行することによって排他処理を開始します。通常、この読み出しに対してはスレーブから EXOKAY 応答が返され、監視対象のアドレスをスレーブが記録したことが通知されます。

注

排他アクセスをサポートしていないスレーブからの排他読み出しをマスタが試行すると、スレーブは EXOKAY 応答ではなく OKAY 応答を返します。マスタはこの応答を、排他アクセスがサポートされていないことを示すエラーとみなすことができます。マスタが排他処理の書き込み部分を実行しないようにすることが推奨されます。

排他読み出しの実行後、マスタは同じ位置への排他書き込みを試行します。この位置が排他読み出しの実行時と同じである場合は、排他書き込み処理が成功します。スレーブによって EXOKAY 応答が返され、排他書き込みによってメモリ位置が更新されます。

このアドレス位置が排他読み出しの実行時と異なる場合には、排他書き込みの試行に失敗し、スレーブによって EXOKAY 応答ではなく OKAY 応答が返されます。排他書き込みが試行されても、このメモリ位置は更新されません。

マスタによって排他処理の書き込み部分が完了されない場合があります。この場合、スレーブは別の排他読み出しによって新しい排他アクセスが開始されるまで、そのアドレスの排他性を監視し続けます。

マスタは、排他アクセスの読み出し部分が完了するまで書き込み部分を開始することはできません。

6.2.3 スレーブの観点から見た排他アクセス

排他アクセスをサポートできる機能を持たないスレーブは、**ARLOCK[1:0]** シグナルと **AWLOCK[1:0]** シグナルを無視できます。スレーブは、通常アクセスと排他アクセスに対して OKAY 応答を返す必要があります。

排他アクセスをサポートしているスレーブは、監視ハードウェアを内蔵している必要があります。このようなスレーブには、スレーブにアクセス可能な排他対応マスタの ID ごとに監視ユニットを組み込むことが推奨されます。単一ポートのスレーブには、標準的な排他アクセスモニタをスレーブの外部に接続できますが、マルチポートのスレーブには内部モニタが必要となる場合があります。

排他アクセスモニタは、排他読み出し処理が行われたアドレスと **ARID** 値を記録します。その後モニタは、そのアドレス位置への書き込みが発生するまで、あるいは同じ **ARID** 値を使用する別の排他読み出しによって異なるアドレスでリセットされるまで、このアドレスを監視し続けます。

ある **AWID** 値を使用して排他書き込みが実行されると、モニタはそのアドレスの排他性が監視中かどうかをチェックします。そのアドレスの排他性が監視中の場合は、その位置への書き込みが発生していないことを意味します。その場合、排他書き込みが継続され、排他アクセスが完了されます。スレーブはマスタに EXOKAY 応答を返します。

排他書き込み時に、そのアドレスが監視対象でない場合、以下のいずれかを意味します。

- 排他読み出しが実行されたあとに、その位置が更新されている。
- モニタに別の位置が再設定されている。

どちらの場合でも、排他書き込みによってそのアドレス位置を更新することはできません。また、スレーブは EXOKAY 応答ではなく OKAY 応答を返します。

6.2.4 排他アクセスの制約条件

排他アクセスには以下の制約条件があります。

- 指定された ID の排他書き込みのサイズおよび長さは、先に行われた同じ ID の排他読み出しのサイズおよび長さと同じである必要があります。
- 排他アクセスのアドレスは、トランザクションの合計バイト数で境界整列されている必要があります。
- 排他読み出しと排他書き込みは同じアドレスである必要があります。
- 排他アクセスの読み出し部分に含まれる **ARID** フィールドは、書き込み部分の **AWID** と一致している必要があります。
- 排他アクセスの読み出し部分と書き込み部分の制御シグナルは同じである必要があります。
- 排他アクセスのバースト転送で転送されるバイト数は、2 の累乗、つまり 1、2、4、8、16、32、64、128 のいずれかである必要があります。
- 排他バーストで転送可能な最大バイト数は 128 バイトです。
- **ARCACHE[3:0]** シグナルと **AWCACHE[3:0]** シグナルには、排他アクセスを監視するスレーブが必ずトランザクションを認識できる値を指定する必要があります。たとえば、スレーブによって監視される排他アクセスで、トランザクションがキャッシュ可能なことを示す値が **ARCACHE[3:0]** あるいは **AWCACHE[3:0]** に指定されてはなりません。

上記の制約条件が守られていない場合の動作は予測不能です。

排他処理の間に監視される最小バイト数は、トランザクションの長さで定義されます。排他アクセスの最大数である 128 バイトまでのデータを監視することができます。しかし、排他アクセスが成功したにも関わらず、近隣のバイトが更新されたために不成功と通知されることがあります。

6.2.5 排他アクセスをサポートしていないスレーブ

応答シグナルの **BRESP[1:0]** と **RRESP[1:0]** には、成功した通常アクセスに対する **OKAY** 応答と、成功した排他アクセスに対する **EXOKAY** 応答が含まれます。つまり、排他アクセスをサポートしていないスレーブは、**OKAY** 応答を返すことによって排他アクセスの失敗を通知することができます。

—— 注 ——

排他アクセスをサポートしていないスレーブへの排他書き込みが実行されると、そのメモリ位置は必ず更新されます。

排他アクセスをサポートしているスレーブへの排他書き込みが実行されると、排他書き込みが成功した場合にのみメモリ位置が更新されます。

6.3 ロックアクセス

トランザクションの **ARLOCK[1:0]** シグナルか **AWLOCK[1:0]** シグナルがロック転送を示している場合、マスタが非ロック転送を完了するまで、そのマスタだけがスレーブの領域にアクセスできることが相互接続によって保障される必要があります。このような状態の確保は相互接続内のアービタによって行われる必要があります。

マスタが、読み出しトランザクションあるいは書き込みトランザクションのロックシーケンスを開始する場合、そのマスタは、完了を待つ他の終了していないトランザクションを持たないことを保証する必要があります。

ARLOCK[1:0] または **AWLOCK[1:0]** がロックシーケンスとして設定されているトランザクションは、相互接続に対して、それに続くトランザクションをロックさせます。そのため、ロックシーケンスは常に、**ARLOCK[1:0]** または **AWLOCK[1:0]** をロックアクセスに設定しない最終トランザクションで完了する必要があります。この最終トランザクションは、ロックシーケンス中に含まれ、実質的にロックの解除を行います。

ロックシーケンスを完了する場合、マスタは、最後のロック解除トランザクションを発行する前に、それまでのすべてのロックトランザクションを完了させる必要があります。その後は、最後のロック解除トランザクションが完全に完了した後、他のトランザクションを開始する必要があります。

マスタは、1 つのロックシーケンス内のすべてのトランザクションに同じ **ARID** 値か **AWID** 値が使用されていることを確認する必要があります。

注

ロックシーケンスの処理中には、相互接続によって他のトランザクションが発生しないことが保証される必要があります。そのため、相互接続の性能に影響する可能性があります。ロックトランザクションは既存のデバイスをサポートする目的でのみ使用することが推奨されます。

以下の制約を適用することが推奨されますが、必須ではありません。

- すべてのロックトランザクションシーケンスを同じ 4KB アドレス領域内で実行する。
- ロックトランザクションシーケンスを 2 つのトランザクションに制限する。

第 7 章

応答シグナル

本章では、AXI の読み出しおよび書き込みトランザクションにおける 4 つのスレーブ
応答について説明します。本章は以下のセクションから構成されています。

- 応答シグナルについて : P. 7-2
- 応答のタイプ : P. 7-4

7.1 応答シグナルについて

AXI プロトコルでは、読み出しトランザクションと書き込みトランザクションの双方に対して応答シグナルを発行できます。読み出しトランザクションの場合、スレーブからの応答情報は、読み出しデータに包含されて渡されますが、書き込みトランザクションの場合には、応答情報は書き込み応答チャネル経由で渡されます。

AXI プロトコルで使用される応答は以下の通りです。

- OKAY
- EXOKAY
- SLVERR
- DECERR

表 7-1 は、**RRESP[1:0]** シグナルと **BRESP[1:0]** シグナルのエンコーディングを示しています。

表 7-1 RRESP[1:0] シグナルと BRESP[1:0] シグナルのエンコーディング

RRESP[1:0] BRESP[1:0]	応答	意味
b00	OKAY	通常アクセスが成功したことを示します。排他アクセスの失敗を示す場合もあります。
b01	EXOKAY	排他アクセスの読み出しあるいは書き込みのいずれかが成功したことを示します。
b10	SLVERR	スレーブのエラーです。アクセスはスレーブに正しく到達したが、スレーブが元のマスタにエラー状態を通知したいときに使われます。
b11	DECERR	デコードのエラーです。通常は、トランザクションアドレスにスレーブが存在しないことを示すために、相互接続コンポーネントによって生成されます。

書き込みトランザクションの場合には、バースト内の各データ転送ごとではなく、バースト全体に対して 1 つの応答があります。

読み出しトランザクションの場合、スレーブはバースト内の転送ごとに異なる応答を返すことができます。たとえば 16 回の読み出し転送が実行されるバーストの場合、スレーブはそのうちの 15 回の転送に対して OKAY 応答を返し、残りの 1 回に対して SLVERR 応答を返すことができます。

このプロトコルによって、エラーが発生した場合にも、要求された回数のデータ転送が行われます。たとえば、マスタによって 8 回の読み出し転送が要求されているときに、スレーブによってエラー条件が検出された場合、スレーブは 8 回の転送を実行し、各転送にエラー応答を返す必要があります。スレーブが 1 つだけエラー応答を返すなら、バーストの残りの部分はキャンセルされません。

このプロトコルでは、複数のアドレスを発行できるマスタには制約条件が課せられ、厳密なエラー通知もサポートしなければなりません。このようなマスタは、後の転送がすでに開始されていても、前の転送に対するエラー応答を処理できなければなりません。

7.2 応答のタイプ

このセクションでは、AXI プロトコルの 4 つの応答タイプについて説明します。

- 通常アクセス成功
- 排他アクセス
- スレーブエラー
- デコードエラー

7.2.1 通常アクセス成功

OKAY 応答は以下を示します。

- 通常アクセスの成功
- 排他アクセスの失敗
- 排他アクセスをサポートしていないスレーブへの排他アクセス

OKAY はほとんどのトランザクションに対する応答です。

7.2.2 排他アクセス

EXOKAY 応答は排他アクセスの成功を示します。第 6 章「アトミックアクセス」でこの応答が説明されています。

7.2.3 スレーブエラー

SLVERR 応答は、トランザクションの失敗を示します。スレーブがエラーとなる条件には以下のようなものがあります。

- FIFO/ バッファのオーバランまたはアンダーランが発生したとき。
- サポートされていないサイズの転送が試行されたとき。
- 読み出し専用位置への書き込みアクセスが試行されたとき。
- スレーブ内でタイムアウトが発生したとき。
- レジスタが存在しないアドレスへのアクセスが試行されたとき。
- ディセーブルまたはパワーダウンされた機能へのアクセスが試行されたとき。

システムの監視とデバッグを容易にするため、エラー応答はエラーが発生した場合にのみ使用し、予想される通常のイベントの通知には使用しないことを推奨します。

7.2.4 デコードエラー

デコードが完全に行われるアドレスマップがないシステムでは、トランザクションに応答するスレーブが存在しないアドレスがある可能性があります。このようなシステムでは、そのアドレスへのアクセスを不正アクセスとしてマークし、存在しないスレーブへのアクセス試行によるシステムのロックを回避するために、相互接続によって適切なエラー応答を返す必要があります。

スレーブへのアクセスが相互接続によって適切にデコードできない場合には、そのアクセスがデフォルトのスレーブにルーティングされ、デフォルトのスレーブによって DECERR 応答が返されます。

エラーがどのように発生したのかを後で参照できるように、デフォルトのスレーブがデコードエラーの情報を記録するように実装できます。この方法で、デフォルトのスレーブによってデバッグプロセスを大幅に簡素化することができます。

AXI プロトコルでは、エラーが発生した場合でも、1 つのトランザクションにおけるデータ転送のすべてを完了することが規定されています。したがって、DECERR 応答を返すコンポーネントは、この要件を満たしている必要があります。

第 8 章

順序モデル

本章では、トランザクション ID タグの使用により、AXI プロトコルによってどのように複数のアドレスを発行し、トランザクションを順不同に処理するのかについて説明します。本章は以下のセクションから構成されています。

- 順序モデルについて : P. 8-2
- 転送 ID フィールド : P. 8-3
- 読み出しの順序 : P. 8-4
- 通常書き込みの順序 : P. 8-5
- 書き込みデータのインタリーブ : P. 8-6
- 読み出しと書き込みの相互作用 : P. 8-8
- 相互接続による ID フィールドの使用 : P. 8-9
- 推奨される ID フィールドの幅 : P. 8-10

8.1 順序モデルについて

AXI プロトコルでは、Out-of-Order トランザクション完了が可能で、複数のアドレスを発行することができます。これらの機能により、高性能な相互接続を実現し、データのスループットとシステムの効率を最大限に高めることができます。

ID シグナルは、順序のある複数のポートとして各ポートを機能させることにより、Out-of-Order トランザクションをサポートします。ある 1 つの ID が付けられたすべてのトランザクションは順序通りに実行される必要がありますが、異なる ID の付いたトランザクションの順序に関する制約はありません。以下の 5 つのトランザクション ID があります。

AWID	書き込みアドレスシグナル群用の ID タグ。
WID	書き込みトランザクション用の ID タグ。マスタは書き込みデータとともに、対応するアドレスの AWID に一致する WID を転送します。
BID	書き込み応答用の ID タグ。スレーブは、スレーブが応答するトランザクションの AWID および WID の値と一致する BID を転送する必要があります。
ARID	読み出しアドレスシグナル群用の ID タグ。
RID	読み出しトランザクション用の ID タグ。スレーブは、スレーブが応答するトランザクションの ARID の値と一致する RID を転送する必要があります。

—— 注 ——

スレーブとマスタがこれらの最新機能を必ず使用しなければならないわけではありません。単純なマスタとスレーブは、発行された順序で一度に 1 つのトランザクションを処理することができます。

複数のアドレスを発行できるということは、マスタは前のトランザクションの完了を待たずともトランザクションアドレスを発行できることを意味します。これによってトランザクションの並列処理が可能となるため、システムのパフォーマンスが向上します。

トランザクションを順序に関係なく実行できるということは、高速メモリ領域へのトランザクションを、それよりも前に開始された遅いメモリ領域へのトランザクションよりも先に完了できることを意味します。これによってトランザクションのレイテンシの影響を軽減できるため、システムのパフォーマンスが向上します。

—— 注 ——

トランザクションの再順序付けは、常に他のトランザクションとの関係を指しています。1 つのバースト内のデータの再順序付けを行う機能はありません。バースト内の転送の順序は、バーストを定義するアドレスシグナルと制御シグナルによって制御されます。

8.2 転送 ID フィールド

AXI プロトコルでは、マスタによって複数の独立したトランザクションを発行できるように ID フィールドが用意されています。トランザクションのそれぞれは順番に返される必要があります。

マスタはトランザクションの **ARID** フィールドか **AWID** フィールドを使用して、マスタの順序の条件に関する付加情報を伝達することができます。トランザクションの順序を決定する規則は以下の通りです。

- 異なるマスタからのトランザクションの順序には制約はありません。これらのトランザクションは任意の順序で実行できます。
- 同一マスタからのトランザクションに異なる ID 値が割り当てられている場合には、順序に関する制約はありません。これらのトランザクションは任意の順序で実行できます。
- 同じ **AWID** 値をもつ一連の書き込みトランザクションのデータは、マスタがそのアドレスを発行した順序と同じ順序で処理される必要があります。
- 同じ **ARID** 値をもつ一連の読み出しトランザクションのデータは、以下のように返される必要があります。
 - 同じ **ARID** の読み出しが同一スレーブから行われる場合には、アドレスを受信した順序と同じ順序で、スレーブによって読み出しデータが返される必要があります。
 - 同じ **ARID** の読み出しが異なるスレーブから行われる場合には、相互接続によって、マスタがアドレスを発行した順序と同じ順序で、読み出しデータが返される必要があります。
- **AWID** と **ARID** が同じ読み出しトランザクションと書き込みトランザクションの間に順序の制約はありません。マスタが順序の制約を必要とする場合、マスタは最初のトランザクションが完全に完了してから次のトランザクションを発行する必要があります。

8.3 読み出しの順序

マスタインタフェースにおいて、同じ **ARID** 値をもつ読み出しトランザクションからの読み出しデータは、マスタがそれらのアドレスを発行した順序と同じ順序で到達する必要があります。**ARID** 値が異なる読み出しトランザクションからのデータは、任意の順序で返すことができます。また、**ARID** フィールド値が異なるトランザクションの読み出しデータをインターリーブすることもできます。

スレーブは、同じ **ARID** 値をもつ一連の読み出しトランザクションからの読み出しデータを、それらのアドレスを受信したときと同じ順序で返す必要があります。異なる **ARID** 値をもつ一連の読み出しトランザクションでは、スレーブはトランザクションが到着したときと異なる順序で読み出しデータを返すことができます。

スレーブは、スレーブが返す読み出しデータの **RID** 値を、応答するアドレスの **ARID** 値と一致させる必要があります。

相互接続は、異なるスレーブからの同じ **ARID** 値をもつ一連の読み出しトランザクションを、順番に完了させる必要があります。

読み出しデータの再順序付けの深さは、再順序付けが可能なスレーブ内に保留されているアドレスの数となります。すべてのトランザクションを順番に処理するスレーブの場合、読み出しデータの再順序付けの深さは「1」です。読み出しデータの再順序付けの深さはスタティックな値であり、スレーブの設計者によって指定される必要があります。

8.4 通常書き込みの順序

スレーブによって書き込みデータのインタリーブがサポートされていない場合には (P. 8-6 「書き込みデータのインタリーブ」を参照)、マスタはトランザクションアドレスの発行時と同じ順序で書き込みトランザクションのデータを発行する必要があります。

スレーブの設計では、書き込みデータのインターリーブをサポートしないのが普通です。そのため、この種のスレーブ設計では、アドレスを受信した順序と同じ順序で書き込みデータを受信する必要があります。異なるマスタからの書き込みトランザクションが相互接続によって 1 つのスレーブに結合される場合には、相互接続はその書き込みデータをアドレスと同じ順序で結合する必要があります。

これらの制約条件は、書き込みトランザクションに異なる **AWID** 値が割り当てられている場合でも適用されます。

8.5 書き込みデータのインタリーブ

書き込みデータのインタリーブ機能により、スレーブインタフェースは異なる **AWID** 値をもつ、インタリーブされた書き込みデータを受けとることができます。スレーブは書き込みデータのインタリーブの深さを宣言することによって、異なる **AWID** 値をもつソースからのインタリーブされた書き込みデータをインタフェースが受けとることができるかどうかを示します。書き込みデータのインタリーブの深さは、スタティックに設定されます。デフォルトでは、どのインタフェースの場合でも、書き込みデータのインタリーブの深さは「1」です。

注

同じ **AWID** を持つ異なるトランザクションの書き込みデータをインタリーブすることはできません。

書き込みデータのインタリーブの深さは、書き込みデータを受信できるスレーブインタフェース内に現在保留しているアドレスの数となります。たとえば、書き込みデータのインタリーブの深さが「2」であり、**AWID** 値がすべて異なる 4 つのアドレスを保留しているスレーブは、最初の 2 つの保留アドレスのどちらかのデータを受信することができます。

スレーブが各トランザクションの最初のデータを受信する順序は、そのトランザクションのアドレスを受信する順序と同じでなくてはなりません。

同一スレーブを対象とする複数の書き込みデータストリームが相互接続によって結合される場合、書き込みデータのインタリーブによって機能停止を回避できる場合があります。相互接続は遅いソースからの 1 つの書き込みデータストリームと、速いソースからの別の書き込みデータストリームを結合する可能性があります。この 2 つの書き込みデータストリームをインタリーブすることで、相互接続はシステムのパフォーマンスを上げることができます。

注

AWID 値が異なる 2 つの書き込みトランザクションによって、同一の、あるいはオーバーラップするアドレス位置がアクセスされる場合の処理順序は定義されていません。より高レベルのプロトコルによって、トランザクションが正しい順序で処理されるようにする必要があります。

1 つの **AWID** 値だけを使用して書き込みデータを生成するマスタインタフェースは、すべての書き込みデータをアドレス発行時と同じ順序で生成します。しかし、スレーブインタフェースの書き込みデータインタリーブの深さが「1」より大きい場合には、マスタインタフェースは異なる **WID** 値のついた書き込みデータをインタリーブすることができます。

書き込みデータの生成を内部制御できるほとんどのマスタでは、書き込みデータのインタリーブを行う必要はありません。このようなマスタは、アドレスを生成した順序と同じ順序で書き込みデータを生成できます。ただし、速度の異なる複数のソースからの書き込みデータを伝送するマスタインタフェースは、それらのソースをインタリーブすることによって相互接続を最大限に活用することができます。

デッドロックを回避するには、インタリーブされた書き込みデータを連続的に受信できる場合にのみ、スレーブインタフェースは書き込み再順序付けの深さを 2 以上に設定する必要があります。スレーブインタフェースは書き込みデータの順序を変更しようとするときに、書き込みデータの受信を停止してはなりません。

8.6 読み出しと書き込みの相互作用

読み出しトランザクションと書き込みトランザクションの間に順序の制約はありません。任意の順序で完了させることができます。

マスタが読み出しトランザクションと書き込みトランザクションの間に順序の制約を必要とする場合、マスタは最初のトランザクションが完全に完了してから次のトランザクションを発行する必要があります。読み出しの場合には、最後の読み出しデータがマスタに返された時点で、前のトランザクションは完了したと見なすことができます。書き込みの場合には、書き込み応答がマスタによって受信された時点でのみ、完了として扱うことができます。すべての書き込みデータが送信されたときに書き込みトランザクションが完了したものとして扱うことはできません。

周辺装置が使用するアドレス領域に関しては、順序の制約が必要な読み出しトランザクションと書き込みトランザクション間を切り替える際に、最初のトランザクションの完了を待機することになります。

メモリ領域に関しては、新しいトランザクションが、同じあるいはオーバーラップしているアドレス領域に対して行われるかどうかを判断するために、トランザクションに対するアドレスのチェック機能をマスタに実装できます。トランザクションがオーバーラップしていない場合には、前のトランザクションの完了を待たずに新しいトランザクションを開始することができます。

8.7 相互接続による ID フィールドの使用

マスタインタフェースが相互接続に接続されている場合、相互接続は **ARID** フィールド、**AWID** フィールド、**WID** フィールドの 3 つのフィールドに、そのマスタポートに固有の追加ビットを付加します。このビットの付加には以下の 2 つの効果があります。

- 相互接続がフィールドにマスタ番号を追加することで固有の ID 値が生成されるため、マスタは他のマスタによって使用される ID 値を知る必要がありません。
- スレーブインタフェース側の ID フィールドの幅が、マスタインタフェース側の ID フィールドよりも広がります。

読み出しデータの場合、相互接続は **RID** フィールドの追加ビットを使用して、その読み出しデータの転送先となるマスタポートを決定します。相互接続は **RID** 値を正しいマスタポートに渡す前に、**RID** フィールドのこれらのビットを削除します。

8.8 推奨される ID フィールドの幅

AXI の Out-of-Order トランザクション機能を活用するため、以下の推奨事項を参考にして下さい。

- マスタコンポーネントには最高 4 ビットまでのトランザクション ID を実装して下さい。
- 相互接続のマスタポート番号に対してはトランザクション ID に 4 ビットまでの追加ビットを実装して下さい。
- スレーブコンポーネントでは 8 ビットの ID を実装して下さい。

—— 注 ——

順序付けされるインタフェースを 1 つだけサポートしたマスタの場合、ID の出力を 0 などの定数値に固定することができます。

順序情報を使用せずにすべてのトランザクションを単に順番に処理するスレーブの場合、標準的な市販のモジュールを使用して、ID の機能をスレーブに付加することができます。そのため、ID の通知機能なしでスレーブの基本的な機能を設計できます。

第 9 章

データバス

本章では、AXI の読み出し / 書き込みデータバスにおける異なるサイズのデータ転送と、AXI がエンディアン混在転送を処理するためにバイト不変エンディアンをどのように使用するかについて説明します。本章は以下のセクションから構成されています。

- データバスについて : P. 9-2
- 書き込みストローブ : P. 9-3
- ナロー転送 : P. 9-4
- バイト不変 : P. 9-5

9.1 データバスについて

AXI プロトコルには 2 つの独立したデータバスがあります。一つは読み出しデータ、もう一つは書き込みデータに使用されます。これらのデータバスにはそれぞれ固有のハンドシェイクシグナルがあるため、両方のバスでデータ転送を同時に発生させることが可能です。

マスタによって生成される各転送の幅は、その転送に使用されるデータバスの幅と同じか、それよりも狭くなくてはなりません。

9.2 書き込みストローブ

書き込みストローブシグナル **WSTRB** によって、書き込みデータベース上で、分散したデータの転送が可能になります。各書き込みストローブシグナルは、書き込みデータベースの 1 バイトに対応しています。このシグナルがアサートされると、書き込みストローブによって、データベースの対応するバイトレーンに、メモリ内で更新される有効な情報があることが通知されます。

書き込みデータベースの 8 ビットごとに 1 つの書き込みストローブがあります。したがって、**WSTRB[n]** は **WDATA[(8 × n) + 7: (8 × n)]** に対応します。図 9-1 は、64 ビットのデータベースにおける関係を示しています。

63	56 55	48 47	40 39	32 31	24 23	16 15	8 7	0
7	6	5	4	3	2	1	0	

図 9-1 バイトレーンのマッピング

マスタは、トランザクションの制御情報から有効だと判断できるデータを保持するバイトレーンに対してのみ、書き込みストローブをアサートする必要があります。

9.3 ナロー転送

マスタによってそのデータベースよりも幅の狭い転送が生成される場合は、転送に使用されるバイトレーンがアドレスと制御情報によって決定されます。インクリメントバーストまたはラッピングバーストでは、バーストの各転送ごとにデータが異なるバイトレーンで転送されます。固定バーストでは一定のアドレスが使用されるため、使用できるバイトレーンも一定です。

図 9-2 と図 9-3 は、バイトレーンの使用例を示しています。

図 9-2 は以下のようになっています。

- バーストが 5 回の転送から構成されています。
- 開始アドレスは 0 です。
- 各転送は 8 ビットです。
- 転送は 32 ビットバス上で発生します。

使用されるバイトレーン			
		DATA[7:0]	1番目の転送
		DATA[15:8]	2番目の転送
	DATA[23:16]		3番目の転送
DATA[31:24]			4番目の転送
		DATA[7:0]	5番目の転送

図 9-2 各転送サイズが 8 ビットのパロー転送

図 9-3 は以下のようになっています。

- バーストが 3 回の転送から構成されています。
- 開始アドレスは 4 です。
- 各転送は 32 ビットです。
- 転送は 64 ビットバス上で発生します。

使用されるバイトレーン	
DATA[63:32]	1番目の転送
	DATA[31:0] 2番目の転送
DATA[63:32]	3番目の転送

図 9-3 各転送サイズが 32 ビットのパロー転送

9.4 バイト不変

同一のメモリ空間内に存在するエンディアン混在のデータ構造体にアクセスする際、AXI プロトコルではバイト不変エンディアン方式が使用されます。

バイト不変エンディアンとは、特定のアドレスへのバイト転送において、同じデータバス線上の 8 ビットのデータが同じアドレス位置に渡されることを意味します。

転送幅が 1 種類しかないコンポーネントでは、そのコンポーネントのバイトレーンをデータバスの適切なバイトレーンに接続する必要があります。複数種類の転送幅をサポートするコンポーネントでは、バイト不変でないインタフェースを変換するためのより複雑なインタフェースが必要となる場合があります。

リトルエンディアンをサポートするコンポーネントのほとんどは、バイト不変インタフェースに直接接続することができます。ビッグエンディアン転送しかサポートしていないコンポーネントは、バイト不変処理を行うための変換機能が必要となります。

図 9-4 は、バイト不変アクセスを必要とするデータ構造の例を示しています。ソース識別子やデスティネーション識別子といったヘッダ情報はリトルエンディアン形式で伝送することが可能ですが、ペイロードはビッグエンディアン形式のバイトストリームで伝送されます。

31	24 23	8 7	0
デスティ-	ソース	パケット	
チェックサム		-ネーション	
ペイロード°		データ項目	
ペイロード°			
ペイロード°			
ペイロード°			

図 9-4 エンディアン混在形式のデータ構造の例

バイト不変方式では、リトルエンディアン形式でヘッダ情報の部分にアクセスした場合でも、そのデータ構造内のビッグエンディアン形式の他のデータは破壊されません。

第 10 章

アンアラインド転送

本章では、アンアラインド転送が AXI プロトコルでどのように処理されるかについて説明します。本章は以下のセクションから構成されています。

- アンアラインド転送について : P. 10-2
- 例 : P. 10-3

10.1 アンアラインド転送について

AXI プロトコルではバースト指向のアドレッシングが使用されるため、各トランザクションでは複数のデータ転送が行われます。通常、各データ転送はその転送サイズに合わせて整列されます。たとえば、32 ビット幅の転送では、4 バイト境界で整列されるのが一般的です。しかし、非境界整列アドレスでバーストを開始した方が望ましい場合があります。

幅が 1 バイトを超えるデータ転送が行われるバーストの場合、アクセスされる必要のある先頭バイトが、本来のデータ幅の境界で整列されていない可能性があります。たとえば、アドレス 0x1002 で始まる 32 ビット（4 バイト）のデータパケットは、32 ビット境界では整列されていません。

AXI プロトコルでは、マスタは下位のアドレスラインを使用して、バーストの非境界整列アドレスを通知できます。下位のアドレスラインに関する情報は、バイトレーンストローブで保持される情報と一貫性が保たれている必要があります。

——— 注 ———

AXI プロトコルでは、マスタからのアライメント情報に基づいてスレーブが特別なアクションを取る必要はありません。

マスタは単に境界整列されたアドレスを伝送すればよく、また書き込みトランザクションでは、データに使用されるバイトレーンに関する情報を伝達するバイトレーンストローブを利用することができます。

10.2 例

図 10-1、P. 10-4 図 10-2、P. 10-4 図 10-3 は、幅の異なるバスにおけるアラインド転送とアンアラインド転送の例を示しています。これらの図の各行は 1 つの転送を表しています。色付きのセルは、アドレスと制御情報に基づいて転送されないバイトを表しています。

	31	24	23	16	15	8	7	0	
	3	2	1	0					1番目の転送
アドレス: 0x00	7	6	5	4					2番目の転送
転送サイズ: 32ビット	B	A	9	8					3番目の転送
バーストタイプ: インクリメント	F	E	D	C					4番目の転送
バーストの長さ: 4転送									
	3	2	1	0					1番目の転送
アドレス: 0x01	7	6	5	4					2番目の転送
転送サイズ: 32ビット	B	A	9	8					3番目の転送
バーストタイプ: インクリメント	F	E	D	C					4番目の転送
バーストの長さ: 4転送									
	3	2	1	0					1番目の転送
アドレス: 0x01	7	6	5	4					2番目の転送
転送サイズ: 32ビット	B	A	9	8					3番目の転送
バーストタイプ: インクリメント	F	E	D	C					4番目の転送
バーストの長さ: 5転送	13	12	11	10					5番目の転送
	7	6	5	4					1番目の転送
アドレス: 0x07	B	A	9	8					2番目の転送
転送サイズ: 32ビット	F	E	D	C					3番目の転送
バーストタイプ: インクリメント	13	12	11	10					4番目の転送
バーストの長さ: 5転送	17	16	15	14					5番目の転送

図 10-1 32 ビットバス上のアラインドおよびアンアラインドワード転送

P. 10-4 図 10-2 は、64 ビットバス上で実行される 32 ビット転送の 3 つのバーストを示しています。

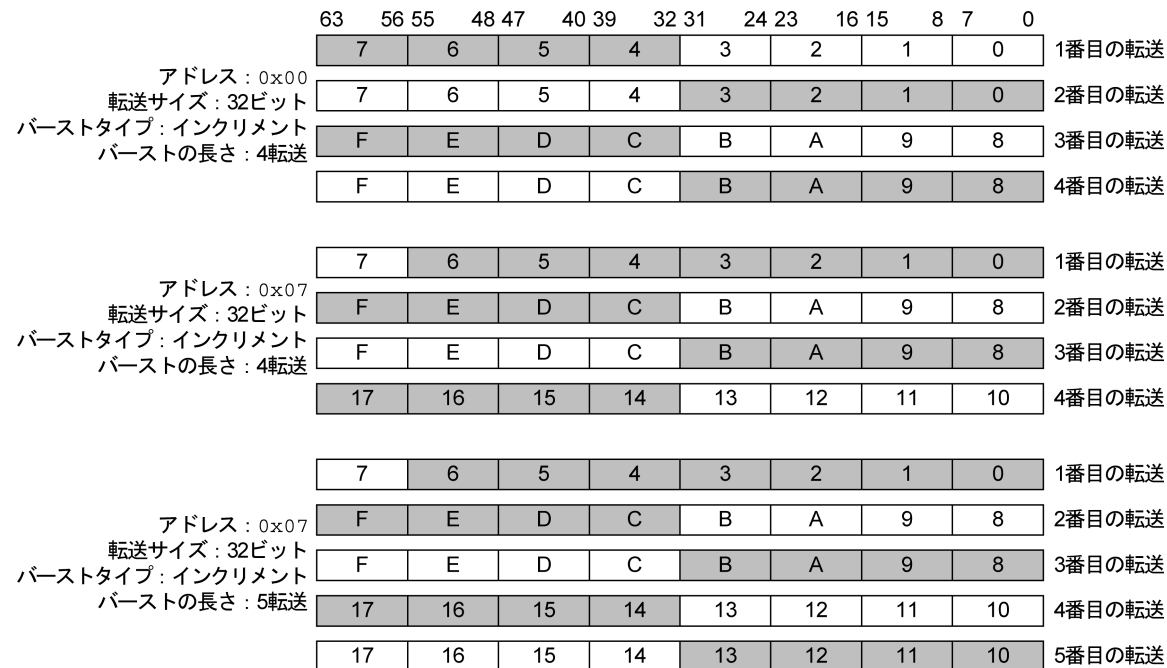


図 10-2 64 ビットバス上のアラインドおよびアンアラインドワード転送

図 10-3 は、64 ビットバス上で実行される 32 ビット転送のラッピングバーストを示しています。

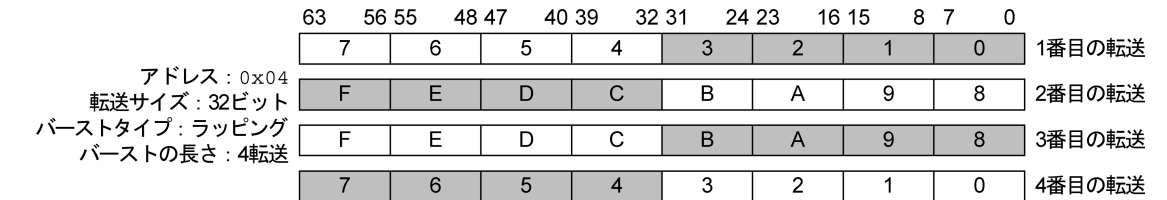


図 10-3 64 ビットバス上のアラインドラッピングワード転送

第 11 章

クロックとリセット

本章では、AXI のクロックシグナルとリセットシグナルのタイミングについて説明します。本章は以下のセクションから構成されています。

- クロックおよびリセットの要件 : P. 11-2

11.1 クロックおよびリセットの要件

このセクションでは、**ACLK** シグナルと **ARESETn** シグナルを実装する際の要件について説明します。

11.1.1 クロック

各 AXI コンポーネントでは、1 つのクロックシグナル **ACLK** が使用されます。すべての入力シグナルは、**ACLK** の立ち上がりエッジでサンプリングされます。すべての出力シグナルの変更は、**ACLK** の立ち上がりエッジの後で発生する必要があります。

マスタとスレーブの両インタフェース上の入力シグナルと出力シグナルの間には組み合わせパスがあつてはなりません。

11.1.2 リセット

AXI プロトコルには、1 つの LOW でアクティブなリセットシグナル **ARESETn** があります。このリセットシグナルは非同期でアサートできますが、アサート解除は **ACLK** の立ち上がりエッジ後に同期して行われる必要があります。

リセット時にはインタフェースに以下のような要件があります。

- マスタインタフェースは、**ARVALID**、**AWVALID**、**WVALID** を LOW にドライブする必要があります。
- スレーブインタフェースは、**RVALID** と **BVALID** を LOW にドライブする必要があります。

その他すべてのシグナルは、任意の値にドライブすることができます。

マスタインタフェースが、**ARVALID**、**AWVALID**、**WVALID** のいずれかを HIGH にドライブできるのは、**ARESETn** が HIGH にセットされた後の **ACLK** の立ち上がりエッジでのみです。図 11-1 は、**ARVALID**、**AWVALID**、**WVALID** を HIGH にドライブできる、リセット後の最初のポイントを示しています。

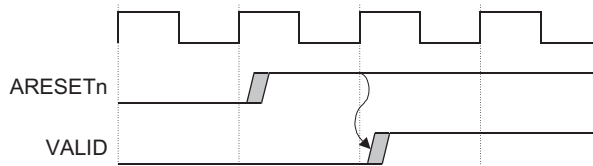


図 11-1 リセット後の終了

第 12 章

低電力インタフェース

本章では、低電力状態の開始から終了までの、AXI プロトコルのクロック制御インタフェースについて説明します。本章は以下のセクションから構成されています。

- 低電力インタフェースについて : P. 12-2
- 低電力クロック制御 : P. 12-3

12.1 低電力インタフェースについて

低電力インタフェースは、データ転送プロトコルのオプションの拡張機能であり、2 種類のペリフェラルを対象としています。

- パワーダウンシーケンスを必要とし、低電力状態に入らなければクロックを停止できないペリフェラル。これらのペリフェラルには、パワーダウンシーケンスをいつ開始するのかを決定するために、システムクロックコントローラからの通知が必要となります。
- パワーダウンシーケンスがなく、クロックをいつ停止できるのかを独自に通知できるペリフェラル

12.2 低電力クロック制御

低電力クロック制御インタフェースには、以下のシグナルがあります。

- クロックのイネーブル/ディセーブルを通知するペリフェラルからのシグナル
- システムクロックコントローラが低電力状態の開始または終了を要求するための 2 つのハンドシェイクシグナル

クロック制御インタフェース内のプライマリシグナルは **CACTIVE** です。ペリフェラルはこのシグナルを使用して、クロックをイネーブルするタイミングを通知します。ペリフェラルは **CACTIVE** をアサートすることによってクロックの要求を通知します。この通知を受けたシステムクロックコントローラはクロックをすぐにイネーブルする必要があります。ペリフェラルは、クロックを必要としないことを通知する場合には **CACTIVE** のアサートを解除します。これによってシステムクロックコントローラは、ペリフェラルのクロックをイネーブルするかディセーブルするかを決定できます。

自らのクロックをいつでもイネーブルまたはディセーブルできるペリフェラルは、**CACTIVE** を常に **LOW** の状態に保つことができます。クロックを常にイネーブルしておく必要のあるペリフェラルは、**CACTIVE** を **HIGH** に固定しておく必要があります。

パワーダウンシーケンスやパワーアップシーケンスのない一部のペリフェラルの場合には、システムクロックコントローラへのこの単純なインタフェースだけで十分です。

パワーダウンシーケンスまたはパワーアップシーケンスを使用する複雑なペリフェラルの場合には、システムクロックコントローラからの要求があった場合にのみ低電力状態に入ります。AXI プロトコルには、この要求をサポートする 2 つの書き込み要求/応答ハンドシェイクシグナルがあります。

CSYSREQ 低電力状態に入るようにペリフェラルに要求する際、システムクロックコントローラは **CSYSREQ** シグナルを **LOW** にセットします。通常動作では、**CSYSREQ** は **HIGH** にセットされます。

CSYSACK ペリフェラルは **CSYSACK** シグナルを使用して、低電力状態開始要求と低電力状態終了要求の両方に応答します。

図 12-1 は、**CSYSREQ** と **CSYSACK** の関係を示しています。

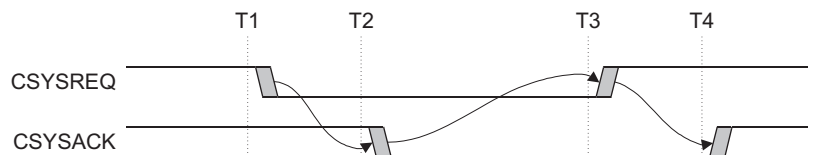


図 12-1 CSYSREQ と CSYSACK のハンドシェイク

P. 12-3 図 12-1 が示すシーケンスの開始時、通常のクロックタイミングで行われる処理の場合には **CSYSREQ** と **CSYSACK** の両方が HIGH にセットされます。T1 でシステムクロックコントローラは **CSYSREQ** をアサート解除し、ペリフェラルに低電力状態に入るよう要求します。ペリフェラルは、T2 で **CSYSACK** をアサート解除することによってこの要求に応答します。T3 でシステムクロックコントローラは **CSYSREQ** をアサートし、低電力状態の終了を通知します。ペリフェラルは T4 で **CSYSACK** をアサートして、その終了要求に応答します。

CSYSREQ と **CSYSACK** のこの関係は、AXI プロトコルで規定されています。

ペリフェラルはシステムクロックコントローラからの低電力状態開始/終了要求を受け入れることも拒否することもできます。ペリフェラルが **CSYSACK** のアサート解除によってシステムクロックコントローラからの要求に応答する際の、**CACTIVE** シグナルのレベルによってその要求の受け入れまたは拒否が通知されます。

12.2.1 低電力要求の受け入れ

図 12-2 は、ペリフェラルがシステムクロックコントローラからの低電力要求を受け入れる際のイベントシーケンスを示しています。

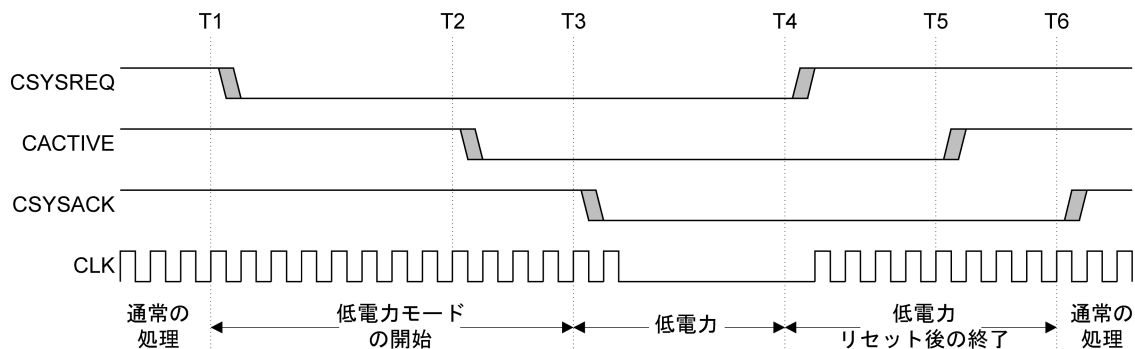


図 12-2 低電力要求の受け入れ

図 12-2 では、システムクロックコントローラが **CSYSREQ** をアサート解除し、ペリフェラルに低電力状態開始を要求する T1 でシーケンスが開始されます。ペリフェラルはこの要求を認識すると、パワーダウン機能を実行して **CACTIVE** をアサート解除できます。次にペリフェラルは T3 で **CSYSACK** をアサート解除し、低電力状態に入ります。

T4 で、システムクロックコントローラは **CSYSREQ** をアサートすることによって低電力状態終了シーケンスを開始します。その後ペリフェラルは T5 で **CACTIVE** をアサートし、さらにその後 **CSYSACK** をアサートすることによって T6 で終了シーケンスを完了します。

12.2.2 低電力要求の拒否

図 12-3 は、ペリフェラルがシステムクロックコントローラからの低電力要求を拒否する際のイベントシーケンスを示しています。

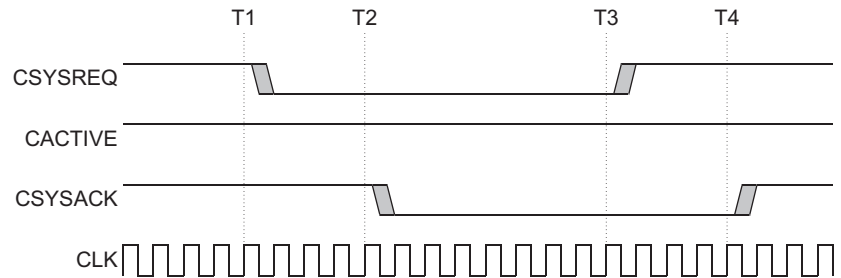


図 12-3 低電力要求の拒否

図 12-3 では、ペリフェラルは低電力要求に応答する際に、**CACTIVE** を HIGH に保持することによって低電力要求を拒否します。その後、システムクロックコントローラは **CSYSREQ** をアサートすることによって低電力要求ハンドシェイクを完了してから別の要求を開始する必要があります。

12.2.3 低電力状態の終了

システムクロックコントローラとペリフェラルのどちらからでも、低電力状態の終了を要求してクロックを復元することができます。低電力状態では **CACTIVE** と **CSYSREQ** の両方が LOW にセットされますが、これらのシグナルのどちらかが HIGH にセットされることによって終了シーケンスが通知されます。

システムクロックコントローラは、クロックをイネーブルし、**CSYSREQ** を HIGH にセットすることによって低電力状態の終了シーケンスを開始できます。次にペリフェラルはパワーアップシーケンスを実行できます。パワーアップシーケンスでは **CACTIVE** が HIGH にセットされます。最後に、**CSYSACK** を HIGH にセットすることによって終了シーケンスが完了します。

ペリフェラルは **CACTIVE** を HIGH にセットすることによって低電力状態の終了シーケンスを開始できます。これを受けてシステムクロックコントローラはすぐにクロックを復元する必要があります。また、**CSYSREQ** を HIGH にセットしてハンドシェイクシーケンスを継続する必要があります。次にペリフェラルは、**CSYSACK** を HIGH にセットすることによってシーケンスを完了し、低電力状態を終了します。ペリフェラルは終了シーケンスを完了するのに必要なサイクル数の間、**CSYSACK** を LOW でホールドできます。

12.2.4 クロック制御シーケンスの概要

図 12-4 は、低電力状態の開始から終了までの典型的なフローを示しています。

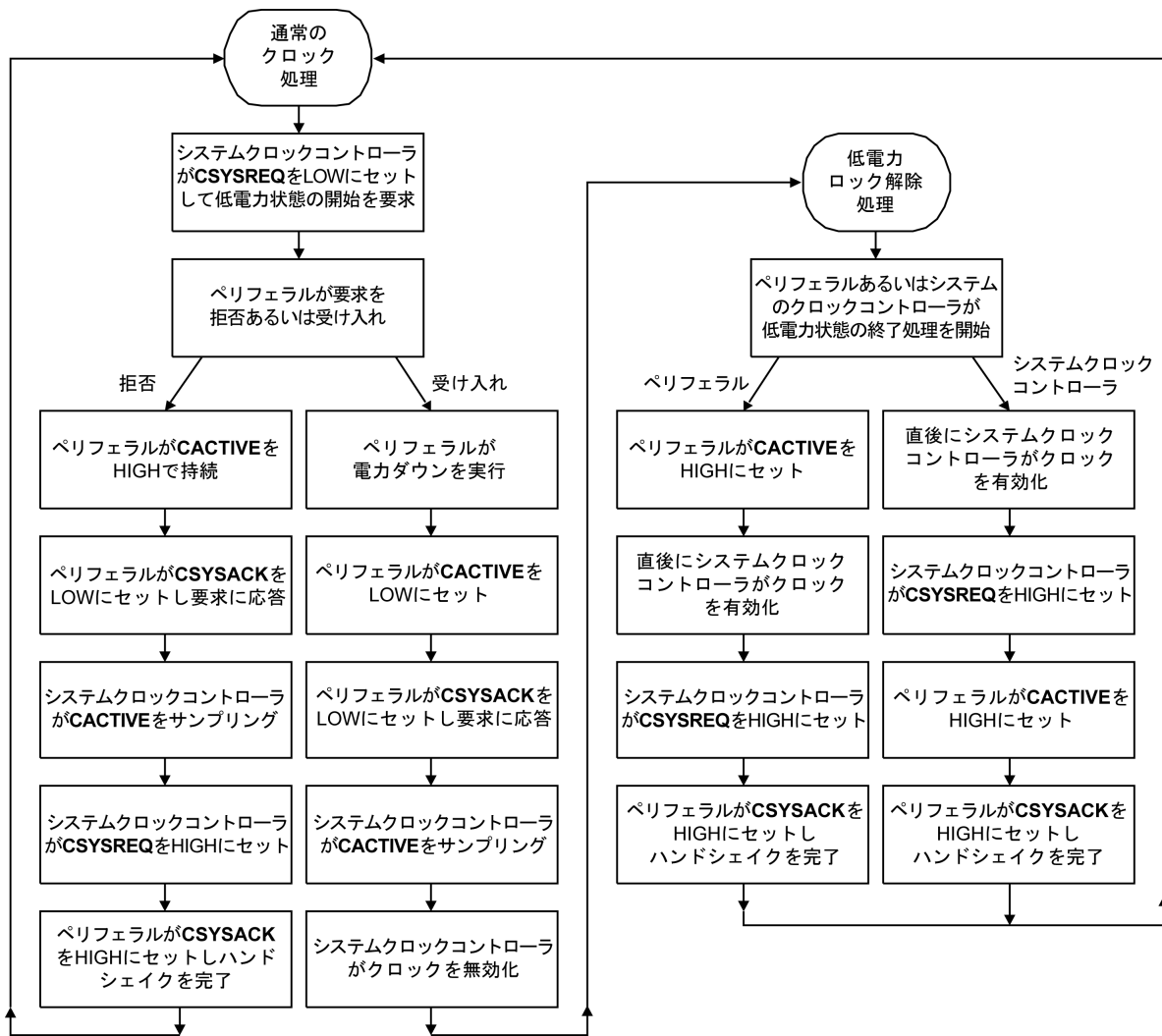


図 12-4 低電力クロック制御シーケンス

12.2.5 低電力ドメインにおけるペリフェラルの結合

システムクロックコントローラは、同じ低電力クロックドメイン内で複数のペリフェラルを結合することができます。以下の規則が守られている場合、このクロックドメインは1つのペリフェラルと同じように処理することができます。

- クロックドメインの **CACTIVE** シグナルは、そのクロックドメイン内のすべての **CACTIVE** シグナルの論理和であること。つまり、システムクロックコントローラは、すべてのペリフェラルがクロックをディセーブルできると通知した場合にのみ、それらのクロックをディセーブルできます。
- システムクロックコントローラは、クロックドメイン内のすべてのペリフェラルにルーティングされた1つの **CSYSREQ** を使用できます。
- クロックドメインの **CSYSACK** シグナルは、以下のように生成されます。
 - **CSYSACK** の立ち下りエッジは、すべてのペリフェラルの最後の立ち下りエッジで発生します。
 - **CSYSACK** の立ち上がりエッジは、すべてのペリフェラルの最後の立ち上がりエッジで発生します。

索引

索引の各項目はアルファベット順に並んでいます。

A

ACLK

説明 2-2

AMBA

アーキテクチャ xiv

インタフェース 1-2

仕様 xvii

ARADDR 2-6

タイミングの例 1-7, 1-8

ARBURST 2-6

エンコーディング 4-5

ARCACHE 2-6

エンコーディング 5-3

排他アクセス 6-5

ARESETn

説明 2-2

タイミング 11-2

ARID 2-6

Out-of-Order トランザクション
8-2

固有性 8-9

排他アクセス 6-4, 6-5

ARLEN 2-6

エンコーディング 4-3

ARLOCK 2-6

エンコーディング 6-2

ARPROT 2-6

エンコーディング 5-6

ARREADY 2-6

タイミングの例 1-7, 1-8

ARSIZE 2-6

エンコーディング 4-4

ARVALID 2-6

タイミングの例 1-7, 1-8

リセット 11-2

AWADDR 2-3

タイミングの例 1-9

AWBURST 2-3

エンコーディング 4-5

AWCACHE 2-3

エンコーディング 5-3

排他アクセス 6-5

AWID 2-3

Out-of-Order トランザクション
8-2

固有性 8-9

排他アクセス 6-5

AWLEN 2-3

エンコーディング 4-3

AWLOCK 2-3

エンコーディング 6-2

AWPROT 2-3

エンコーディング 5-6

AWREADY 2-3

タイミングの例 1-9

AWSIZE 2-3

エンコーディング 4-4

AWVALID 2-3

タイミングの例 1-9

リセット 11-2

AXI プロトコル

機能 1-2

B

- BID 2-5
 - Out-of-Order トランザクション 8-2
- BREADY 2-5
 - タイミングの例 1-9
 - デフォルト値 3-4
- BRESP 2-5
 - エンコーディング 7-2
 - タイミングの例 1-9
 - 排他アクセス 6-6
- BVALID 2-5
 - タイミングの例 1-9
 - リセット要件 11-2

C

- CACTIVE 2-8
 - タイミングの例 12-4, 12-5
- CSYSACK 2-8
 - タイミングの例 12-4, 12-5
- CSYSREQ 2-8
 - タイミングの例 12-4, 12-5

D

- DECERR 応答 7-2, 7-5
- DMA, サポート 1-2

E

- EXOKAY 応答 6-3, 6-5, 7-2, 7-4

O

- OKAY 応答 6-4, 6-5, 7-2, 7-4
- Out-of-Order トランザクション 1-3, 1-9
 - 1 マスタから 8-3
 - 書き込み順序の深さ 8-6
 - 相互接続 1-9, 8-2

R

- RDATA 2-7
 - タイミングの例 1-7, 1-8
- RID 2-7
 - Out-of-Order トランザクション 8-2
 - 固有性 8-9
- RLAST 2-7
 - タイミングの例 1-7, 1-8
- RREADY 2-7
 - タイミングの例 1-7, 1-8
 - デフォルト値 3-5
- RRESP 2-7
 - エンコーディング 7-2
 - 排他アクセス 6-6
- RVALID 2-7
 - タイミングの例 1-7, 1-8
 - リセット要件 11-2

S

- SLVERR 応答 7-2, 7-4

W

- WDATA 2-4
 - タイミングの例 1-9
- WID 2-4
 - Out-of-Order トランザクション 8-2
 - 固有性 8-9
- WLAST 2-4
 - タイミングの例 1-9
- WREADY 2-4
 - タイミングの例 1-9
- WSTRB 2-4
 - バイトレーンのマッピング 9-3
- WVALID 2-4
 - タイミングの例 1-9
 - リセット要件 11-2

ア

- アトミックアクセスのエンコーディング 6-2

- アドレッシングオプション 4-2
- アドレス ID タグ
 - ARID および AWID 参照
- アドレスチャネル 1-3
 - 定義 1-4
 - ハンドシェイク 1-4, 3-2
- アンアラインド転送 10-1
 - アンアラインド開始アドレスの通知 10-2
 - 例 10-3, 10-4
- インクリメントバースト
 - インクリメント値 4-5
 - 開始アドレス 4-7
 - ナロー 4-4
 - バイトレーン 4-4, 4-8
- インタリーブされたトランザクション
 - Out-of-Order トランザクション参照
- 応答シグナル 7-2

力

- 書き込み ID タグ
 - WID 参照
- 書き込みアドレスチャネルシグナル 2-3
 - ハンドシェイク 3-3
- 書き込み応答チャネル 1-3
 - シグナル 2-5
 - 定義 1-5
 - ハンドシェイク 1-4, 3-2, 3-4
- 書き込みストローブシグナルWSTRB 参照
- 書き込みデータチャネル 1-3
 - シグナル 2-4
 - 定義 1-5
 - バイトレーンストローブ 1-5
 - ハンドシェイク 1-4, 3-2, 3-4
- 書き込みデータのインタリーブの深さ 8-6
- 書き込みトランザクション
 - 完了通知 1-5
- 書き込み割り当て属性
 - 選択 5-2
- 仮想マスタ 1-10
- 完了通知 1-5, 7-2, 7-4, 7-5
 - BRESP 参照
 - RRESP 参照

規則

シグナルの命名 xvii

タイミング図 xvi

表記規則 xv

キャッシュ

サポート 5-2

キャッシュ可能属性 1-11

選択 5-2

キャッシュのエンコーディング 5-3

グローバルシグナル 2-2

クロック 11-2

固定バースト 4-5

開始アドレス 4-7

バイトレーン 4-4, 4-8

サ

シグナル

書き込みアドレスチャネル 2-3

書き込み応答チャネル 2-5

書き込みデータチャネル 2-4

グローバル 2-2

低電力インタフェース 2-8

読み出しアドレスチャネル 2-6

読み出しデータチャネル 2-7

シグナルの命名規則 xvii

スレーブエラー応答

SLVERR 応答参照

相互接続

Out-of-Order トランザクション
1-9, 8-6アドレスとデータのリアライニ
ング 3-6

実装 1-6

データストリームの結合 8-6

ロックアクセス 6-7

タ

タイミング図の規則 xvi

ダイレクトメモリアクセス

DMA 参照

チャネルレジスタの挿入 1-6

通常アクセスの成功

OKAY 応答参照

低電力インタフェース

シグナル 2-8

データバス

ナロー転送 9-4

幅 1-5

デコードエラー

DECERR 応答参照

トランザクション ID タグ 1-9, 1-10

ARID および AWID 参照

RID 参照

WID 参照

相互接続 8-9

トランザクション属性 1-11

トランザクションの順序

Out-of-Order トランザクション参
照

書き込み順序の深さ 8-6

書き込みデータのインタリーブ
8-6

書き込みトランザクション 8-5

規則 8-3

読み出しトランザクション 8-4

ハ

バースト

アドレス 4-7

長さ 4-7

バーストサイズ 4-4

エンコーディング 4-4

バーストタイプ 4-5

インクリメント 4-5

エンコーディング 4-5

固定 4-5

ラッピング 4-6

バースト長 4-3

エンコーディング 4-3

バーストのラッピング

開始アドレス 4-6, 4-7

長さ 4-3, 4-6

ナロー 4-4

バイトレーン 4-4, 4-8

ラップ境界 4-6, 4-7

排他アクセス

スレーブサポートロジック 6-3

選択 6-2

排他アクセス応答

EXOKAY 応答参照

バイト不変エンディアン 9-5

バイトレーン

32 ビット転送の例 9-4

8 ビット転送の例 9-4

バイトレーンストローブ 1-5

WSTRB 参照

バッファリング可能属性 1-11

選択 5-2

ハンドシェイク

アドレスチャネル 3-2

書き込みアドレスチャネル 3-3

書き込み応答チャネル 3-2, 3-4

書き込みデータチャネル 3-2,
3-4

シグナルの依存性 3-7

タイミングの例 3-3

読み出しアドレスチャネル 3-4

読み出しデータチャネル 3-2,
3-5ビッグエンディアン of データ構造
9-5

表記規則 xv

並列トランザクション処理 1-6,
1-8, 8-2

ペリフェラルクロック制御 12-3

保護

エンコーディング 5-6

保護レベル

選択 5-5

マ

マスタ - スレーブハンドシェイク
1-4, 1-7, 3-2

タイミングの例 3-2, 3-3

ヤ

読み出し ID タグ

RID 参照

読み出しアドレスチャネル

シグナル 2-6

ハンドシェイク 3-4

読み出しデータチャネル 1-3

シグナル 2-7

定義 1-5

ハンドシェイク 1-4, 3-2, 3-5

読み出し割り当て属性

選択 5-2

ラ

リセット 11-2

リトルエンディアンのデータ構成
9-5

レジスタの挿入 1-6

ロックアクセス

相互接続 6-7

選択 6-2

ワ

割り当て属性 1-11