AMBA™ 仕様書

(Rev 2.0)

ARM

ARM IHI 0011AJ-00

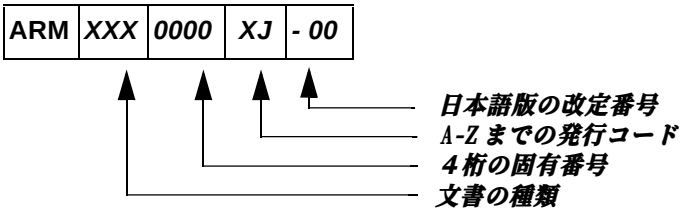
AMBA 仕様書
(Rev 2.0)

Release information

キー

文書番号

本書には識別するための番号が固有に付いています。この番号は、表紙及び各ページ下部に記載されています。



文書ステータス

文書ステータスは各ページ下部の見出しに記載されています。
これによって、文書の機密性及び記載されている情報の状況を知ることができます。

以下は機密性の状況を示します。

ARM Confidential	ARM 社員及び NDA 署名者のみ閲覧可。
Named Partner Confidential	上記及び指定提携企業の社員のみ閲覧可。
Partner Confidential	ARM 社員及び全提携企業の社員のみ閲覧可。
Open Access	制約無し。

以下は情報の状況を示します。

Advance	将来の製品に関する情報
Preliminary	開発中の製品に関する最新情報
Final	完成品に関する最終情報

本書の改訂履歴は以下の通りです。

改訂履歴

日付	発行	改訂
1999/5/13	A	初版

日本語版

発行	日付	改訂
AJ-00	1999 年 10 月	初版

Proprietary notice

ARM、ARM Powered ロゴ、Thumb、StrongARM は、ARM Limited の登録商標です。

ARM ロゴ、AMBA、PrimeCell、Angel、ARMulator、EmbeddedICE、ModelGen、Multi-ICE、ARM7TDMI、ARM7TDMI-S、ARM9TDMI、TDMI、STRONG は、ARM Limited の商標です。

本書に記載されているその他全ての製品またはサービス名は各社の商標です。

本書に記載されている情報および製品の全部または一部について、著作権保有者の文書による事前の許可を得ない限り、転用あるいは複製することはできません。

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 Limited が利用者の利益のためだけに提供するものです。したがって当社では、製品の市販性または利用の適切性を含め、暗示的、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または損傷についても、ARM Limited は一切責任を負いません。

本書は製品を使用する際の補助的なものとして書かれたものです。本製品をご使用になる場合は、英語版をご利用ください。

Document confidentiality status

本書は、オープンアクセス扱いで、配布に関していかなる規制もありません。

Product status

本書に記載されている情報は、（開発製品に関する）最終的な情報です。

ARM web address

`http://www.arm.com`

序文

序文では、*アドバンスドマイクロコントローラバスアーキテクチャ* (AMBA) の仕様書について説明しており、以下のセクションから構成されています。

本書について

本書は、AMBA 仕様書です。

対象とする読者

本書は、AMBA 仕様書に適合するモジュールを設計する経験豊かなハードウェア技術者およびソフトウェア技術者を対象としています。

構成

本書は、以下の章から構成されています。

Chapter 1 *AMBA バスの概要*

AMBA バスについて概要を説明します。

Chapter 2 *AMBA 信号*

各 AMBA 装置で使用する信号について説明します。

Chapter 3 *AMBA AHB*

AMBA アドバンスドハイパフォーマンスバスについて説明します。

Chapter 4 *AMBA ASB*

AMBA アドバンスドシステムバスについて説明します。

Chapter 5 *AMBA APB*

AMBA アドバンスドペリフェラルバスについて説明します。

Chapter 6 *AMBA 試験方法*

AMBA バスで用いる試験方法について説明します。

表記規則

本書では以下の表記規則を使用しています。

bold
テキスト内の ARM プロセッサ信号の名前、およびメニューの名前などのインターフェース要素を表しています。また、必要に応じて記述リスト内の強調箇所にも使用されています。

italic
専門語、相互参照、引用事項を表しています。

`typewriter`
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で入力できるテキストを表しています。

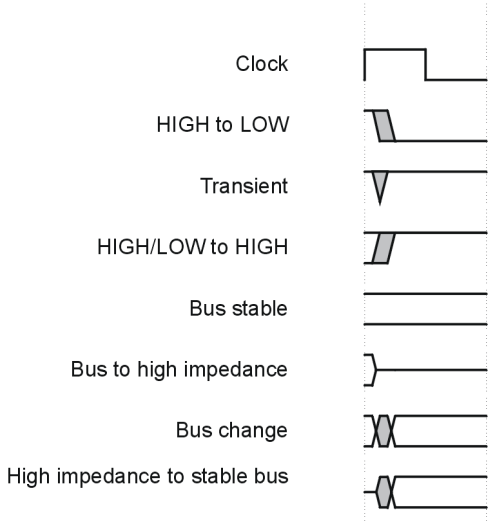
`typewriter`
コマンドまたはオプションに使用できる省略語を表しています。コマンドまたはオプション名を全て入力する代わりに、下線部の文字だけを入力することができます。

`typewriter italic`
特定の値に置き換えられる、コマンドおよび関数の引数を表しています。

`typewriter bold`
外部の例示コードを使用したときの言語キーワードを表しています。

タイミングダイアグラム規則

本書では、1 つ以上のタイミングダイアグラムを使用しています。これらのタイミングダイアグラムで使用するキーコンポーネントを以下に示します。なお、これらのコンポーネントのバリエーションを使用するときには、その旨明示します。したがって、具体的に示していない限り、別の意味を表わすことはありません。



タイミングダイアグラム規則の標記

バスおよび信号の斜線部は不定状態を示しているので、そのバスや信号は、そのときの斜線部内の任意の値をとることができます。この値の実レベルは重要なものではなく、正常動作には影響を及ぼしません。

お問い合わせ

ARM Limited では、AMBA 製品および AMBA 仕様書に関する皆様からのご意見・ご質問をお待ちしております。

本書に関するご意見・ご質問

本書に関してのご意見・ご質問等がありましたら、以下の情報と共に errata@arm.com まで電子メールをお送り下さい。

- 資料の題名
- 資料番号
- ご意見・ご質問のあるページ番号
- お問い合わせ内容の詳しい説明

本書に盛り込んで欲しい内容や改善すべき点などもお聞かせ下さい。

AMBA 仕様書に関するご意見・ご質問

本製品に関するご意見・ご質問等がある場合は、本製品をご購入になった販売店にお問い合わせ下さい。その際、以下の情報をご提供下さい。

- 製品名
- お問い合わせ内容の詳しい説明

目次
AMBA 仕様書

Chapter 1
AMBA バスの概要

本章では、*アドバンスド マイクロコントローラバスアーキテクチャ* (AMBA) 仕様書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。

1.1 AMBA 仕様書の概要

アドバンスドマイクロコントローラバスアーキテクチャ (AMBA) 仕様書は、高性能組み込み型マイクロコントローラを設計するためのオンチップ通信基準を規定します。

AMBA 仕様書内では、異なった 3 種類のバスが定義されています。

- *アドバンスドハイパフォーマンスバス* (AHB)
- *アドバンスドシステムバス* (ASB)
- *アドバンスドペリフェラルバス* (APB)

試験方法は、モジュラーマクロセル試験や診断アクセスのインフラストラクチャを提供する AMBA 仕様書に付属しています。

1.1.1 アドバンスドハイパフォーマンスバス(AHB)

AMBA AHB は高性能・高クロック周波数システムモジュール用です。

AHB は高性能システムの*中枢*バスとして機能します。AHB はプロセッサ、オンチップメモリ、および低電力周辺マクロセル機能を持ったオフチップ外部メモリインターフェースの接続を効率化します。また、AHB は合成と自動試験手法を用いた効率的な設計フローにおいて使いやすいように規定されています。

1.1.2 アドバンスドシステムバス (ASB)

AMBA ASB は高性能システムモジュール用です。

AMBA ASB とは高性能な機能を持ったAHBが必要とされない場合に合わせた代替のシステムバスです。また、ASB はプロセッサ、オンチップメモリ、および低電力周辺ユニットのマクロセル機能を持ったオフチップ外部メモリインターフェースの接続を効率化します。

1.1.3 アドバンスドペリフェラルバス (APB)

AMBA APB は低電力周辺ユニット用です。

AMBA APB は周辺ユニットの機能をサポートするために、電力消費をできるだけ小さくし、インターフェースの複雑さを低減するように最適化されています。APB は両方のバージョンのシステムバスに連結して使用できます。

1.2 AMBA 仕様書の目的

AMBA 仕様書の作成目的は、以下の 4 つの要求を満たすことです。

- 1つ以上の CPU またはシグナルプロセッサを搭載する組み込み型マイクロコントローラ製品の適切かつ迅速な開発を容易にする。
- *特殊な技術に依存せず*、周辺ユニットの再利用性を高め、システムのマクロセルが多様なプロセスの範囲にまたがって移植出来るようになっており、かつフルカスタムやスタンダード・セル、ゲートアレイの技術にも適していること。
- プロセッサの独立性を向上するために*モジュラーシステム設計*を促進し、先進のキャッシュ付き CPU コアの開発ロードマップを提供したり、周辺ユニットのライブラリを開発する。
- 動作と製造試験の両者のためのオンチップ通信とオフチップ通信を効率化するのに必要とされるシリコンの基本設計手法を極めて簡単にする。

1.3 代表的な AMBA ベースのマイクロコントローラ

AMBA ベースのマイクロコントローラは通常、外部メモリ周波数帯域を維持できる高性能システムの*中枢*バス (AMBA AHB、または AMBA ASB) から構成されています。このバスには CPU、オンチップメモリ、その他の*ダイレクトメモリアクセス* (DMA) ユニットが接続されています。このバスは転送の大部分に関与する素子間の高周波数帯域用インターフェースを提供します。また、この高性能バスには、システム内のほとんどの周辺ユニットが接続され (図 1-1 参照)、低周波数帯域用 APB へのブリッジも接続されています。

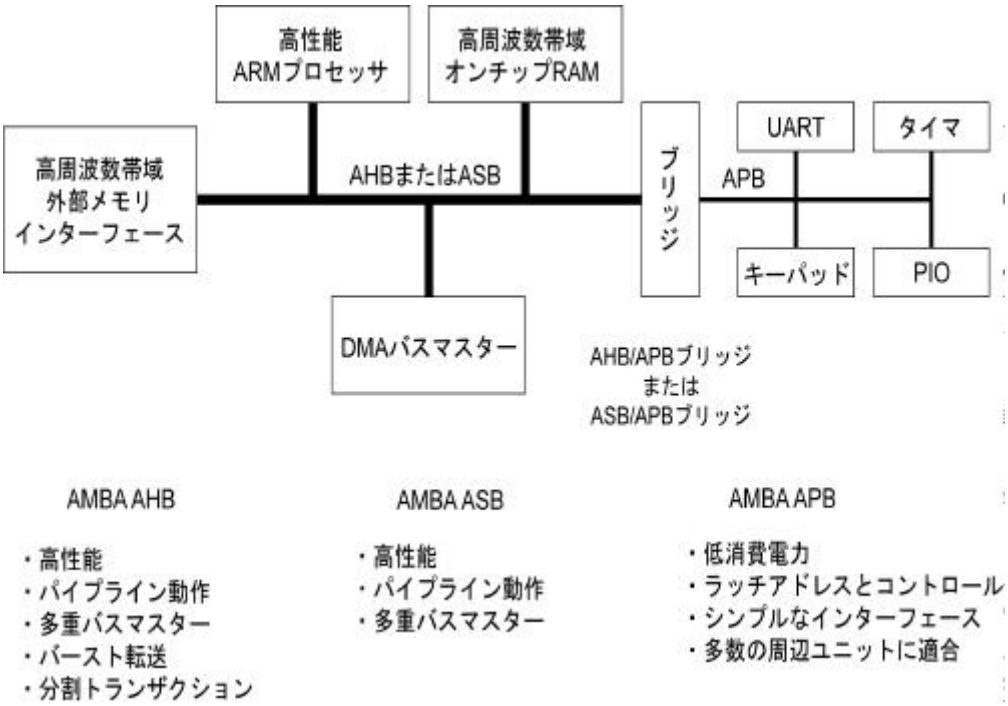


図 1-1 代表的な AMBA システム

AMBA APB は、周波数帯域がより高いパイプライン方式メインシステムバスからの二次バスとして、周辺マクロセル通信の基本設計手法を提供します。そのような周辺ユニットは通常、以下のような特徴があります。

- メモリマップレジスタとなるインターフェースを備えています。
- 高周波数帯域用インターフェースは持っていません。
- プログラム制御のもとでアクセスされます。

外部メモリインターフェースはその用途に特化されており、狭いデータ経路しか持っていない場合があります。しかし、システムに独立した試験装置を用いて、内部の AMBA AHB、ASB、および APB モジュールを個別に試験できるテストアクセスモードをサポートしている場合もあります。

1.4 専門用語集

本仕様書では一貫して、以下の用語を使用します。

バスサイクル バスサイクルは 1 回のバスクロック時間の基本単位であり、AMBA AHB または APB プロトコルの記述のために、立上がりエッジから立上がりエッジまでの区間と定義されています。ASB バスサイクルは立下がりエッジから立下がりエッジまでの区間と定義されています。バス信号タイミングはバスサイクルクロックを基準としています。

バス転送 AMBA ASB または AHB バス転送はデータの読出しまたは書込み動作のことで、場合により 1 サイクル以上のバスサイクルが必要です。このバス転送は、アドレス指定されたスレーブから完了応答があると終了します。

AMBA ASB がサポートしている転送サイズには、バイト (8 ビット)、ハーフワード (16 ビット)、ワード (32 ビット) があります。AMBA AHB はさらに幅広いデータ転送をサポートしており、64 ビット転送と 128 ビット転送もサポートしています。AMBA APB バス転送はデータの読出しまたは書込み動作のことで、常に 2 バスサイクルが必要です。

バースト動作 バースト動作は、バスマスターが起動する 1 回以上のデータトランザクションと定義され、アドレス空間の増分領域に合わせて一定のトランザクション幅を持っています。トランザクション 1 回当たりの増分は転送 (バイト、ハーフワード、ワード) の幅によって決まります。APB 上でのバースト動作はサポートされていません。

1.5 AMBA AHB の概要

AHB は次世代の AMBA バスで、高性能で論理合成可能な設計の要求に応えます。AHB は複数のバスマスターをサポートし、高周波数帯域動作を可能にする高性能システムバスです。

AMBA AHB は高性能・高クロック周波数システムに必要な機能を実装しており、以下の機能が含まれています。

- バースト転送
- 分割トランザクション
- シングルサイクルバスマスターハンドオーバー
- シングルクロックエッジ動作
- 非トライステート実行
- 広幅データバス構成 (64/128 ビット)

この高度なバスと現行の ASB/APB はブリッジで容易に接続でき、これにより既存の設計を容易に統合できます。

AMBA AHB 設計では 1 つ以上のバスマスターを組み込むことができ、システムは通常、最低でもプロセッサとテストインターフェースを含みます。しかし、*ダイレクトメモリアクセス* (DMA) または *ディジタル信号プロセッサ* (DSP) をバスマスターとした組み込みが一般に行なわれています。

外部メモリインターフェース、APB ブリッジ、および内部メモリは最も一般的な AHB スレーブです。システム内のその他の周辺ユニットも AHB スレーブとして組み込むことができます。しかし、低周波数帯域用周辺ユニットは通常、APB に接続されます。

代表的な AMBA AHB システム設計では、以下のコンポーネントが組み込まれています。

AHB マスター バスマスターは、アドレスと制御情報を提供することにより読出し動作や書込み動作を開始できます。バスを実際に使用できるのは一度に 1 つのバスマスターだけです。

AHB スレーブ バススレーブは、一定範囲のアドレス空間内での読出し動作や書込み動作に応答します。バススレーブはアクティブマスターにデータ転送の成功、失敗、または待ちの信号を返します。

AHB アービタ バスアービタは、一度に 1 つのバスマスターだけにデータ転送を開始させます。たとえアービトレーションプロトコルが固定されていても、アプリケーションの要求により、*最優先*アクセスまたは*正当*アクセスなどのアービトレーションアルゴリズムを実行できます。
単一バスマスターシステムにおいては、どうでもよいことかもしれませんが、AHB にはアービタが 1 つしか組み込まれません。

AHB デコーダ　AHB デコーダは、各転送のアドレスを解読し、その転送に関与するスレーブの選択信号を提供します。
すべての AHB に対して 1 つの集中型デコーダが必要です。

1.6 AMBA ASB の概要

ASB は第一世代の AMBA システムバスです。ASB は現行の APB の上位に位置し、以下の機能を含む、高性能システムに必要な機能を実装しています。

- バースト転送
- パイプライン転送動作
- マルチバスマスター

代表的な AMBA ASB システムは、1 つ以上のバスマスター、例えば、最低でもプロセッサとテストインターフェースを組み込めます。しかし、*ダイレクトメモリアクセス* (DMA) または *ディジタル信号プロセッサ* (DSP) をバスマスターとして組み込みが一般に行なわれています。

外部メモリインターフェース、APB ブリッジ、および内部メモリは最も一般的な ASB スレーブです。システム内のその他の周辺ユニットも ASB スレーブとして組み込むことができます。しかし、低周波数帯域用周辺ユニットは通常、APB に接続されます。

AMBA ASB システム設計では通常、以下のコンポーネントが組み込まれています。

ASB マスター バスマスターは、アドレスと制御情報を提供することにより読出し動作や書込み動作を開始できます。バスを実際に使用できるのは一度に 1 つのバスマスターだけです。

ASB スレーブ バススレーブは、一定範囲のアドレス空間内での読出し動作や書込み動作に応答します。バススレーブはアクティブマスターにデータ転送の成功、失敗、または待ちの信号を返します。

ASB デコーダ バスデコーダは転送アドレスを解読し、適切なスレーブを選択します。また、バス転送が必要でないときにもバスを動作可能状態に保持します。
すべての AHB に対して 1 つの集中型デコーダが必要です。

ASB アービタ バスアービタは、一度に 1 つのバスマスターだけにデータ転送を開始させます。たとえアービトレーションプロトコルが固定されていても、アプリケーションの要求により、*最優先*アクセスまたは*正当*アクセスなどのアービトレーションアルゴリズムを実行できます。
単一バスマスターシステムにおいては、どうでもよいことかもしれませんが、ASB にはアービタが 1 つしか組み込まれません。

1.7 AMBA APB の概要

APB は AMBA バス階層の一部で、消費電力を最少にし、インターフェースの複雑さを低減するために最適化されています。

AMBA APB は、単一の AHB または ASB スレーブデバイスとしてまとめられたローカル二次バスとして機能します。APB は AHB または ASB 信号に直接基づいた低消費電力な拡張機能をシステムバスに提供します。

APB ブリッジはローカル周辺バスの代わりにバスハンドシェイクと制御信号の再度タイミング調節を処理するスレーブモジュールとして機能します。システムバスの起点から APB インターフェースを規定することにより、システム診断と試験方法の利点を利用できます。

AMBA APB は、周波数帯域が小さく、パイプライン方式バスインターフェースに高性能を必要としない周辺ユニットとのインターフェースをとるために使用すべきです。

最新バージョンの APB は、すべての信号遷移がクロックの立上がりエッジによってのみ決まるように規定されています。この改良により、APB 周辺ユニットをどんな設計フローにも容易に統合でき、以下の利点が得られます。

- 高周波数動作をより容易に実現できます。
- 性能がクロックのマーク／スペース比に依存していません。
- シングルクロックエッジの使用により静的タイミング解析が簡単になりました。
- 自動テスト挿入について特別に考慮する必要がありません。
- 多くの *特定用途向け IC* (ASIC) ライブラリには、より優れた立上がりエッジレジスタが選択されています。
- サイクルベースなシミュレータとの統合が容易です。

APB に対するこれらの変更により、APB と AHB 間の接続もより容易になります。

AMBA APB の実装には通常、AHB または ASB 転送を ASB 上のスレーブデバイスに適したフォーマットの変換に必要な 1 つの APB ブリッジが組み込まれます。このブリッジは、APB 周辺ユニットの選択信号を発生させるのに、解読するだけでなく、アドレス信号、データ信号、および制御信号のすべてをラッチします。

APB に接続されているその他のモジュールはすべて APB スレーブです。APB スレーブは以下のインターフェース仕様に基づいています。

- アドレス信号と制御信号は、アクセスの間はたえず有効です（非パイプライン）。

- **非周辺バスがアクティブの時、無電力消費のインターフェースです (周辺バスは未使用時、完全停止です)。**
- **タイミングは、ストローブ信号タイミングでデコードすることによって提供できます (ロックされていないインターフェース)。**
- **書込みデータは全アクセスに有効です (明確なラッチ実行によってグリッチ除去を可能にします)。**

1.8 システムに適したバスの選択

システムで使用するバスを決定する前に、以下の項目について検討すべきです。

- *システムバスの選択*
- *システムバスおよび周辺バス*
- *AMBA AHB/ASB または APB を使用すべき場合*：P.1-13

1.8.1 システムバスの選択

AMBA AHB と ASB は両者ともメインシステムバスとして使用可能です。システムバスの選択は通常、必要なシステムモジュールが提供するインターフェースによって決まります。

AHB は、すべての新しい設計に推奨します。AHB がより広い周波数帯域に対応しているという理由だけでなく、シングルクロックエッジプロトコルによって、典型的な ASIC 開発時に使用される設計自動化ツールとの統合がよりスムースになるという理由からです。

1.8.2 システムバスおよび周辺バス

すべての周辺ユニットに全機能型 AHB または ASB モジュールとして構築することは、実現可能ですが、いつも望ましいというわけではありません。

- 周辺マクロセルを多く使用した設計では、バスの負荷が大きくなり、電力損失を増大させ、性能を犠牲にします。
- タイミング解析が必要な場合、バス上の最も遅い素子により最高性能が制限されます。
- パイプライン方式信号から恩恵を受ける高周波数帯域マクロセルとは対照的に、多くの単純周辺マクロセルはラッチされたアドレスと制御信号を必要とします。
- 多くの周辺ユニットの機能は、マクロセル選択と読出し / 書込みバス動作を伝達する選択ストローブを必要とするだけです。この際、高周波数クロック信号を各周辺ユニットに送信する必要はありません。

1.8.3 AMBA AHB/ASB または APB を使用すべき場合

完全な AHB または ASB インターフェースは以下の要素に使用します。

- バスマスター
- オンチップメモリブロック
- 外部メモリインターフェース
- FIFO インターフェースを用いた高周波数帯域周辺ユニット
- DMA スレーブ周辺ユニット

単純 APB インターフェースは以下の要素に推奨します。

- 単純レジスターマップスレーブデバイス
- クロックを全体に送信できない、極低電力インターフェース
- システムバスへの負荷を回避するために行なう狭幅バス周辺ユニットのグループ分け

1.9 AMBA 仕様書に関する留意事項

本 AMBA 仕様書を読む際には以下の点について考慮して下さい。

- *技術からの独立性*
- *電気的特性*
- *タイミングの仕様*

1.9.1 技術からの独立性

AMBA は技術から独立しているオンチッププロトコルです。本仕様書はクロックサイクルレベルでバスプロトコルを詳述しているだけです。

1.9.2 電気的特性

電気的特性に関する情報は、設計に選択された製造プロセス技術に全て依存するので、AMBA 仕様書には記載されていません。

1.9.3 タイミングの仕様

AMBA プロトコルは各種信号の挙動をサイクルレベルで規定しています。正確なタイミングに対する要求は、使用されるプロセス技術と動作周波数に左右されます。

厳密なタイミングの必要条件は AMBA プロトコルによって定義されていないので、バス上の各種モジュール間で信号タイミングを割り付ける際、システムインテグレータには最大の自由度が与えられます。

Chapter 2
AMBA 信号

本章では、AMBA 信号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。

2.1 AMBA 信号の名前

AMBA 信号はすべて、名前の第一文字目がその信号が関係するバスの種類を示すように名付けられています。信号名の小文字 **n** は、その信号がアクティブローであることを示し、そうでない場合、信号名はすべて大文字になります。

テスト信号は、バスの種類にかかわらず、プレフィクス **T** をつけます。テスト信号についての詳細は、第 6 章「*AMBA 試験方法*」を参照して下さい。

2.1.1 AHB 信号のプレフィクス

H AHB 信号を示します。

例えば、**HREADY** は AHB 転送のデータ部分が完了できることを示すために使用します。この信号はアクティブハイです。

2.1.2 ASB 信号のプレフィクス

A ASB バスマスターとアービタ間の一方向信号です。

B ASB 信号です。

D 一方向 ASB デコーダ信号です。

例えば、**BnRES** は ASB リセット信号です。この信号はアクティブローです。

2.1.3 APB 信号のプレフィクス

P APB 信号を示します。.

例えば、**PCLK** は APB が使用するメイン・クロックです。

2.2 AMBA AHB 信号リスト

本セクションでは、AMBA AHB 信号の概要を示します (表 2-1 参照)。各信号の詳細については、本書の後ろのセクションを参照して下さい。

この種類の信号はすべて、プレフィクス H が付いており、システム設計において AHB 信号と、似た名前の他の信号とを区別しています。

表 2-1 AMBA AHB 信号

名前	信号源	説明
HCLK バスクロック	クロック源	このクロックはすべてのバス転送のタイミングをとります。信号タイミングはすべて、HCLK の立上がりエッジによって決まります。
HRESETn リセット	リセットコントローラ	バスリセット信号はアクティブローで、システムとバスのリセットに使用します。これは唯一のアクティブロー信号です。
HADDR[31:0] アドレスバス	マスター	32 ビットシステムのアドレスバスです。
HTRANS[1:0] 転送タイプ	マスター	現在の転送タイプを示します。可能な転送タイプは NONSEQUENTIAL、SEQUENTIAL、IDLE、または BUSY です。
HWRITE 転送方向	マスター	この信号は、HIGH のとき、書込み転送を示し、LOW のとき読出し転送を示します。
HSIZE[2:0] 転送サイズ	マスター	転送のサイズを示し、通常、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) です。プロトコルは最大で 1024 ビットまでの転送サイズを可能にします。
HBURST[2:0] バーストの種類	マスター	転送がバーストの一部を発生させたかどうかを示します。4、8、16 ビートのバーストがサポートされ、バーストはインクリメント式かラップ式のどちらかです。
HPROT[3:0] 保護制御	マスター	プロテクション制御信号はバスアクセスに関する追加情報を提供し、プロテクションのいくつかのレベルを実現したいモジュールによって本来使用されるものです。 この信号は転送が特権モードアクセスか、ユーザーモードアクセスかだけでなく、オペコード取出しか、データアクセスかも示します。メモリ管理ユニット付きのバスマスターの場合、この信号は現在のアクセスがキャッシュ可能かバッファ可能かも示します。

表 2-1 AMBA AHB 信号（続き）

名前	信号源	説明
HWDATA[31:0] 書込みデータバス	マスター	書込みデータバスは、書込み動作時、マスターからバススレーブにデータを転送するために使用します。32 ビット以上のデータバス幅を推奨します。しかし、これはより高い周波数帯域での動作に対処するため容易に拡張できます。
HSELx スレーブ選択	デコーダ	各 AHB スレーブは独自のスレーブ選択信号を持ち、この信号は現在の転送が選択したスレーブ向けであることを示します。この信号は単に、アドレスバスの組合わせデコードです。
HRDATA[31:0] 読込みデータバス	スレーブ	読出しデータバスは、読出し動作時、バススレーブからバスマスターにデータを転送するために使用します。32 ビット以上のデータバス幅を推奨します。しかし、これはより高い周波数帯域での動作に対処するため容易に拡張できます。
HREADY 転送完	スレーブ	**HREADY** 信号は、HIGH の場合、バスでの転送が終了したことを示します。この信号は転送を延ばすために LOW になる場合があります。 注：バス上のスレーブは、入力と出力の両方の信号として **HREADY** を要求とします。
HRESP[1:0] 転送応答	スレーブ	転送応答は転送状態に関する追加情報を提供します。 以下の 4 つの応答があります。OKAY、ERROR、RETRY、および SPLIT。

また、AMBA AHB はマルチバスマスター動作をサポートするのに必要な数多くの信号を持っています (表 2-2 参照)。これらのアービトレーション信号の多くは専用ポイント・ツー・ポイントリンクです。表 2-2 で、サフィクス x はモジュール X から送信されたこと示します。例えば、1 つのシステム内で、HBUSREQarm、HBUSREQdma、HBUSREQtic など、多くの HBUSREQx 信号があります。

表 2-2 アービトレーション信号

名前	信号源	説明
HBUSREQx バス要求	マスター	バスマスター x からバスアービタへの信号で、そのバスマスターがバスを要求していることを示します。システム内の最大 16 個のバスマスターのそれぞれに HBUSREQx 信号があります。
HLOCKx ロックされた転送	マスター	この信号は、HIGH のとき、マスターがロックされたバスアクセスを要求していることを示し、この信号が LOW になるまで、バスの使用を他のマスターに対して許可しません。
HGRANTx バス認可	アービタ	この信号はバスマスター x が現在、最優先マスターであることを示します。HREADY が HIGH のとき、アドレス / 制御信号の所有権は転送終了時に移るので、HREADY と HGRANTx の両者が HIGH になると、別のマスターがバスへアクセスできます。
HMASTER[3:0] マスター番号	アービタ	アービタからの本信号は、どのバスマスターが現在、転送を行なっているかを示します。SPLIT 転送をサポートしているスレーブが使用して、どのマスターがアクセスを試みているかを判断します。 HMASTER のタイミングはアドレスと制御信号のタイミングに一致しています。
HMASTLOCK ロックされたシーケンス	アービタ	現在のマスターがロックされた一連の転送を実行していることを示します。この信号のタイミングは HMASTER 信号のタイミングと同じです。
HSPLITx[15:0] 分割完了要求	スレーブ (SPLIT 可能)	この 16 ビット分割バスは、スレーブが使用して、どのバスマスターに分割トランザクションの再試行を許可すべきかをアービタに知らせます。 この分割バスの各ビットは 1 つのバスマスターに対応しています。

2.3 AMBA ASB 信号リスト

表 2-3 に AMBA ASB 信号を示しています。

表 2-3 AMBA ASB 信号

名前	説明
AGNTx バス認可	バスアービタからバスマスター x への信号で、**BWAIT** が LOW のとき、そのバスマスターがバスを使用できることを示します。システムの各バスマスターには、関連バス要求信号 **AREQx** のみならず、**AGNTx** 信号もあります。
AREQx バス要求	バスマスター x からバスアービタへの信号で、そのバスマスターがバスを要求していることを示します。システムの各バスマスターには、関連バス要求信号 **AGNTx** のみならず、**AREQx** 信号もあります。
BA[31:0] アドレスバス	システムアドレスバスで、アクティブバスマスターによって駆動されます。
BCLK バスクロック	このクロックがすべてのバス転送のタイミングをとります。バスでの転送を制御するために、**BCLK** の LOW 状態と HIGH 状態が使用されます。
BD[31:0] データバス	双方向システムデータバスです。このデータバスは、書込み転送時には現在のバスマスターによって駆動され、読出し転送時には選択されたバススレーブによって駆動されます。
BERROR エラー応答	転送エラーは、選択されたバススレーブが **BERROR** 信号を使って知らせます。**BERROR** が HIGH のときは、転送エラーが発生したことを、**BERROR** が LOW のときは、転送が成功したことを意味します。また、この信号は **BLAST** 信号と共に使用されてバスリトラクト動作を示します。 スレーブが選択されていない場合、この信号はバスデコーダによって駆動されます。
BLAST 最終応答	この信号は、選択されたバススレーブによって駆動され、現在の転送がバーストシーケンスの最後かどうかを示します。 **BLAST** が HIGH のとき、デコーダはアドレスの解読に十分な時間を確保する必要があります。**BLAST** が LOW のとき、次の転送がバーストシーケンスを続行できます。この信号は **BERROR** 信号と組み合わせて使用して、バスリトラクト動作を示します。 スレーブが選択されていない場合、この信号はバスデコーダによって駆動されます。
BLOK ロックされた転送	この信号は、HIGH のとき、現在の転送と次の転送が分割できず、その他のバスマスターがバスにアクセスできないことを示します。この信号はバスアービタが使用します。 この信号はアクティブバスマスターによって駆動されます。
BnRES リセット	バスリセット信号はアクティブローで、システムとバスをリセットするために使用します。この信号は唯一のアクティブロー信号です。

表 2-3 AMBA ASB 信号（続き）

名前	説明
BPROT[1:0] 保護制御	プロテクション制御信号は、バスアクセスに関する追加情報を提供し、主に基本的なプロテクション・ユニットとして機能する時にバスデコーダが使用するためのものです。この信号は転送が特権モードアクセスか、ユーザーモードアクセスかだけでなく、オペコード取出しか、データアクセスかを示します。この信号は、アクティブバスマスターによって駆動され、そのタイミングはアドレスバスのタイミングと同じです。
BSIZE[1:0] 転送サイズ	転送サイズ信号は転送のサイズを示します。可能なサイズはバイト、ハーフワード、ワードです。 この信号は、アクティブバスマスターによって駆動され、そのタイミングはアドレスバスのタイミングと同じです。
BTRAN[1:0] 転送タイプ	この信号は次のトランザクションの転送タイプを示します。可能な転送タイプは ADDRESS-ONLY、NONSEQUENTIAL、または SEQUENTIAL です。この信号は、該当する **AGNTx** 信号がアクティブになると、バスマスターによって駆動されます。
BWAIT 待ち応答	この信号は選択されたバススレーブによって駆動され、現在の転送を完了してもよいかどうかを示します。**BWAIT** が HIGH のときは、さらなるバスサイクルが必要です。**BWAIT** が LOW のときは、現在のバスサイクルで現在の転送を完了できます。 スレーブが選択されていない場合、この信号はバスデコーダによって駆動されます。
BWRITE 転送方向	この信号は、HIGH のとき、書込み転送を示し、LOW のとき読出し転送を示します。この信号は、アクティブバスマスターによって駆動され、そのタイミングはアドレスバスのタイミングと同じです。
DSELx スレーブ選択	バスデコーダからバススレーブへの信号で、そのスレーブデバイスが選択されており、データ転送が必要であることを示します。各 ASB バススレーブ用に **DSELx** 信号があります。

2.4 AMBA APB 信号リスト

すべての AMBA APB 信号が 1 文字のプレフィクス **P** を使用します。クロックなどの一部の APB 信号は、システムバスの同等信号に直接接続できます。

表 2-4 に、AMBA APB 信号名リストを示し、各信号の使用方法も付記します。

表 2-4 AMBA APB 信号

名前	説明
PCLK バスクロック	APB 上の全転送のタイミングをとるために、**PCLK** の立上がりエッジを使用します。
PRESETn APB リセット	APB バスリセット信号はアクティブローで、この信号は通常、システムバスリセット信号に直接接続します。
PADDR[31:0] APB アドレスバス	この信号は APB アドレスバスです。バス幅が最大 32 ビットで、周辺バスブリッジユニットによって駆動されます。
PSELx APB 選択	周辺バスブリッジユニット内の二次デコーダから各周辺バススレーブ x への信号です。本信号は、そのスレーブデバイスが選択されており、データ転送が必要であることを示します。各バススレーブ用に **PSELx** 信号があります。
PENABLE APB ストローブ	このストローブ信号は、周辺バス上の全アクセスのタイミングをとります。イネーブル信号は APB 転送の 2 番目のサイクルを示すために使用します。**PENABLE** の立上がりエッジは APB 転送の途中で現われます。
PWRITE APB 転送方向	この信号は、HIGH のとき、APB 書込みアクセスを示し、LOW のとき読出しアクセスを示します。
PRDATA APB 読出しデータバス	読出しデータバスは、読出しサイクル時 (**PWRITE** が LOW のとき)、選択されたスレーブによって駆動されます。そのバス幅は最大 32 ビットです。
PWDATA APB 書込みデータバス	書込みデータバスは、書込みサイクル時 (**PWRITE** が HIGH のとき)、周辺バスブリッジユニットによって駆動されます。そのバス幅は最大 32 ビットです。

Chapter 3
AMBA AHB

本章では、*アドバンスドハイパフォーマンスバス* (AHB) アーキテクチャ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。

3.1 AMBA AHB について

AHB は次世代の AMBA バスで、高性能で合成可能な設計の要求に応えます。AMBA AHB は、APB の上位に位置している新しいレベルのバスで、高性能・高クロック周波数システムに必要な機能を実装しており、以下の機能が含まれています。

- バースト転送
- 分割トランザクション
- シングルサイクルバスマスターハンドオーバー
- シングルクロックエッジ動作
- 非トライステート実行
- 広幅データバス構成 (64/128 ビット)

3.1.1 代表的な AMBA AHB ベースマイクロコントローラ

AMBA ベースマイクロコントローラは通常、外部メモリ周波数帯域を維持できる高性能システム*中枢*バスから構成されています。このバスには CPU とその他の*ダイレクトメモリアクセス* (DMA) デバイスが接続されています。さらに、より低い周波数帯域の周辺ユニットが配置されているより狭幅 APB バスへのブリッジも接続されています。図 3-1 に、代表的な AMBA システムの AHB と APB を示します。

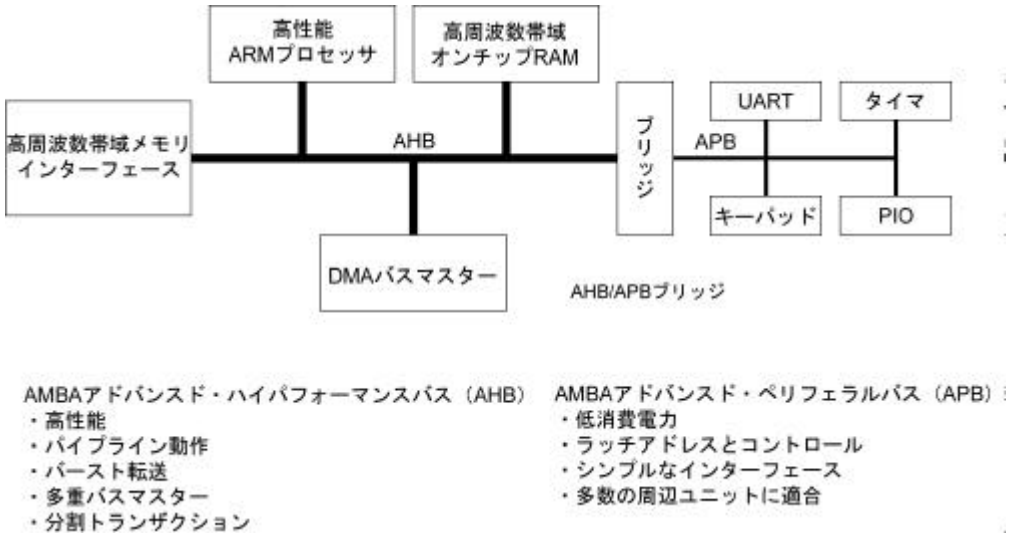


図 3-1 代表的な AMBA AHB ベースシステム

3.2 バス相互接続

AMBA AHB バスプロトコルは、セントラルマルチプレクサ相互接続方式で使用するように設計されている。この方式を使用して、すべてのバスマスターは、実行したい転送を示すアドレスおよび制御信号を送信し、アービタはどのマスターがアドレスと制御信号を全スレーブに送信させるかを決めます。また、セントラルデコーダは読出しデータ・応答信号マルチプレクサを制御するために必要なもので、転送に関与するスレーブからの適切な信号を選択します。

図 3-2 に、3 台のマスターと 4 台のスレーブを使用した AMBA AHB 設計を実現するのに必要な構成を示します。

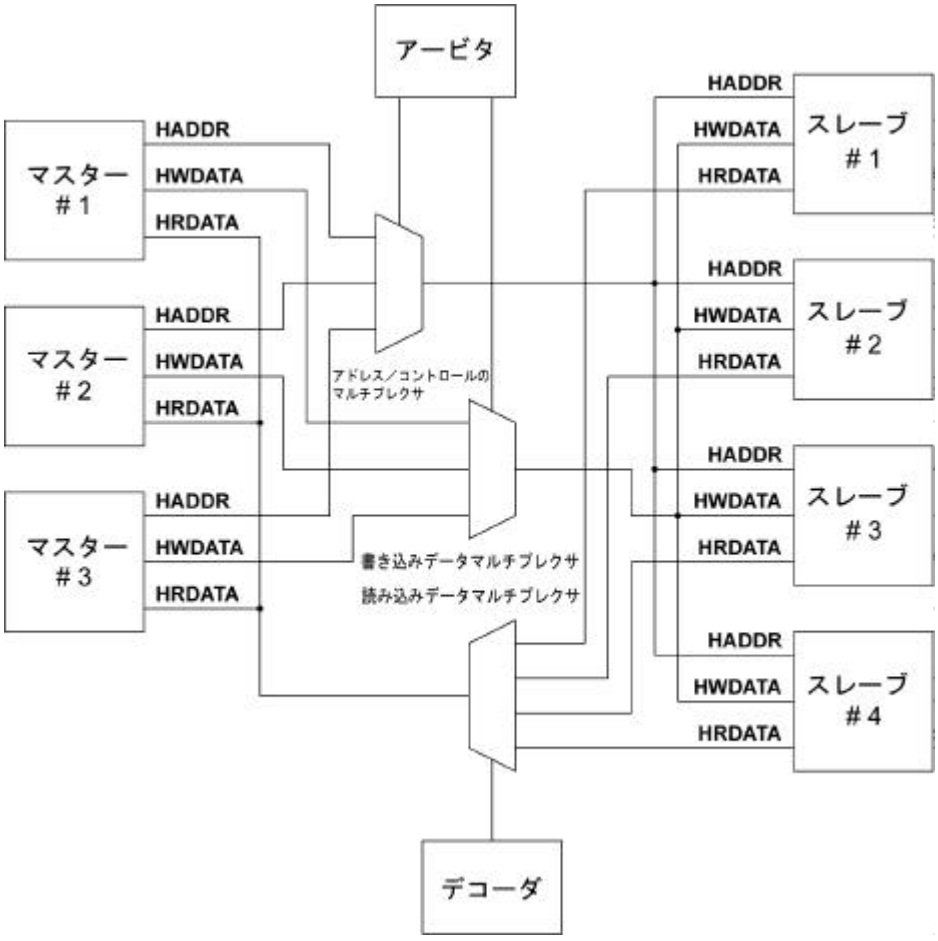


図 3-2 マルチプレクサ相互接続

3.3 AMBA AHB 動作の概要

AMBA AHB 転送の開始が可能となる前に、バスマスターがバスへのアクセスを認可しなければなりません。このプロセスは、マスターがアービタへの要求信号をアクティブにすることにより開始します。その後、アービタはいつマスターにバスの使用が認可されるかを知らせます。

認可されたバスマスターは、アドレスと制御信号を駆動することによって AMBA AHB 転送を開始します。これらの信号は、その転送がバーストの一部を発生させるかどうかを示すだけでなく、その転送のアドレス、方向、幅に関する情報も提供します。以下の 2 つの異なった形態のバースト転送が可能です。

- インクリメント式バースト。これはアドレス境界でラップしません。
- 折返し式バースト。これは特定のアドレス境界でラップします。

書込みデータバスはマスターからスレーブへデータを移動し、読出しデータバスはスレーブからマスターへデータを移動するために使用します。

各転送は以下のサイクルから構成されます。

- アドレスと制御サイクル
- データ用の 1 つ以上のサイクル

アドレスは拡張できないので、すべてのスレーブがこの間にアドレスを取得する必要があります。しかし、データは **HREADY** 信号を使用して拡張できます。この信号が LOW のとき、ウェイトステートを転送中に挿入し、スレーブに対し、データを供給したり取得したりするための余分な時間を与えます。

転送時、スレーブはその状態を応答信号 **HRESP[1:0]** を使用して知らせます。

OKAY	OKAY 応答は、その転送が正常に進行していることを示すために使用され、**HREADY** が HIGH になると、この信号はその転送が成功の上で完了したことを示します。
ERROR	ERROR 応答は、転送エラーが発生し、その転送が失敗したことを示します。
RETRY と SPLIT	RETRY と SPLIT 転送応答は両者とも、その転送がすぐには完了できないが、バスマスターはその転送を試み続けるべきであることを示します。

正常動作では、アービタが別のマスターにバスへのアクセスを認可する前に、マスターが特定バーストで全転送を完了するのを認めます。しかし、過大なアービトレーション待ち時間を回避するために、アービタにバースト中止させることが可能です。そのような場合、そのバースト内の残りの転送を完了させるために、マスターがバスアービトレーションを再度行なう必要があります。

3.4 基本転送

AHB 転送は以下の 2 つの異なったセクションから構成されます。

- アドレスフェーズ。これはほんの 1 サイクル持続するだけです。
- データフェーズ。これは場合により、数サイクル必要です。**HREADY** 信号を使用して実現します。

図 3-3 に、最も単純な転送、すなわちウェイトステートがない転送を示します。

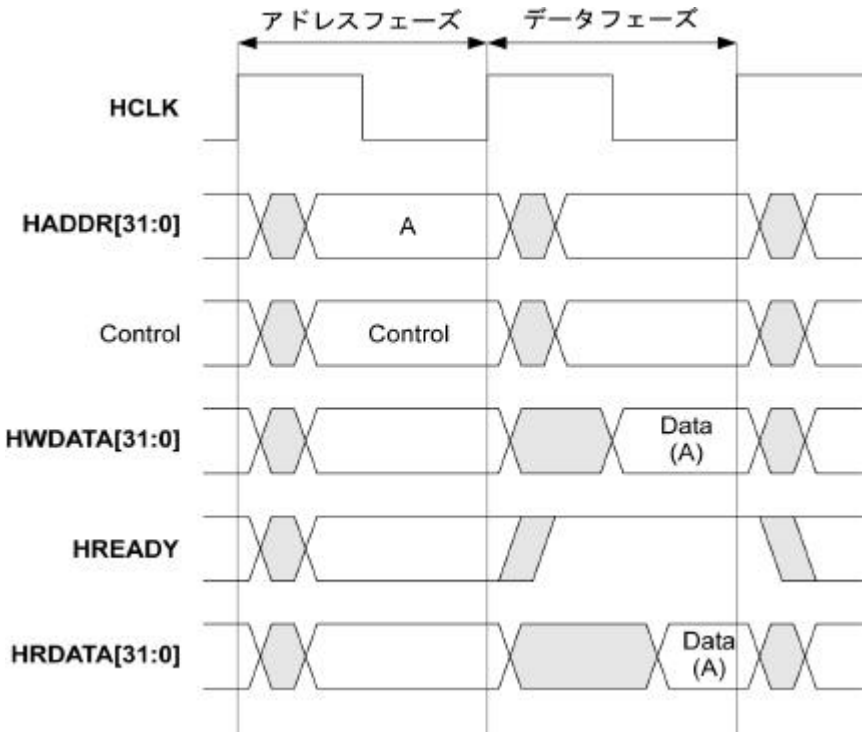


図 3-3 単純な転送

ウェイトステートがない単純な転送では、

- マスターは、**HCLK** の立上がりエッジの後、アドレスと制御信号をバス上に送出します。
- スレーブは、その後、クロックの次の立上がりエッジ時にアドレスと制御情報を取得します。

- アドレスと制御の取得後、スレーブは適切な応答を駆動し始めることができ、この応答はクロックの 3 番目の立上がりエッジ時にバスマスターによって取得されます。

この簡単な例では、転送のアドレスフェーズとデータフェーズが互いに異なったクロック時間にどのように発生するかを説明します。実は、転送のアドレスフェーズは直前の転送のデータフェーズ中に発生します。アドレスとデータのこのオーバーラップは、バスのパイプライン処理の性質の根底をなすもので、高性能動作に対処しながら、他方ではスレーブは転送に応答する十分な時間を確保します。

図 3-4 に示すように、スレーブは転送中にウェイトステートを挿入することができ、これにより、その転送が延長され、完了するための時間が確保できます。

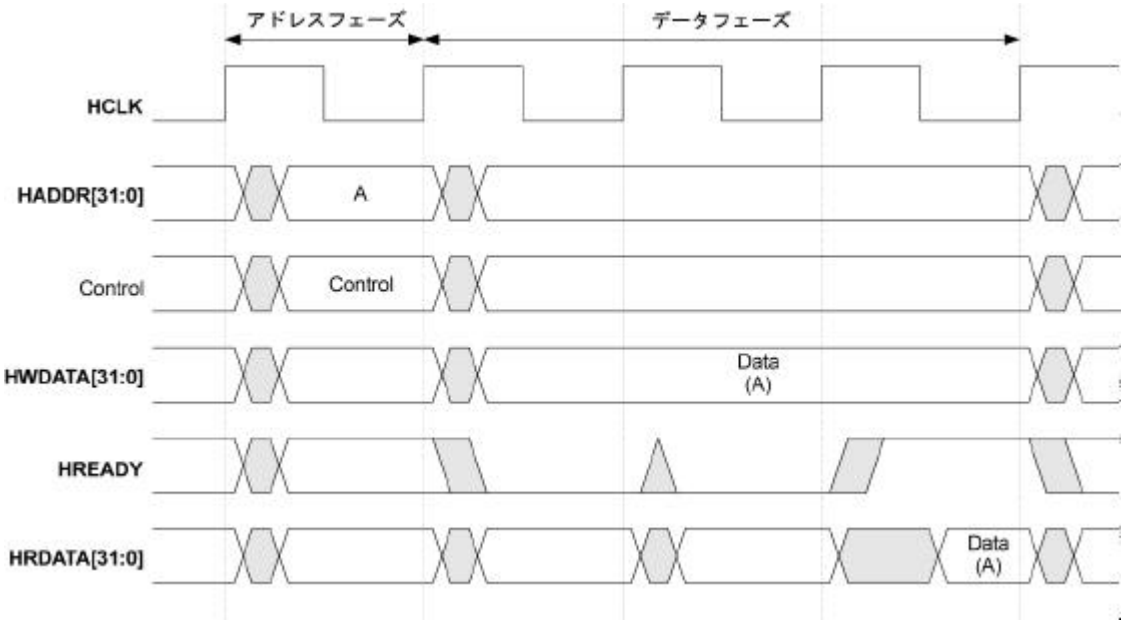


図 3-4 ウェイトステートを挿入した転送

Note

書込み動作の場合、バスマスターは延長サイクルの間ずっとデータを安定状態に保持します。

読出し転送の場合、転送が完了するまで、スレーブが有効データを供給する必要はありません。

このように転送を延長すると、次の転送のアドレスフェーズが延びるという副作用が生じます。これを図 3-5 に、互いに無関係なアドレス A、B、C への 3 つの転送を使用して示しています。

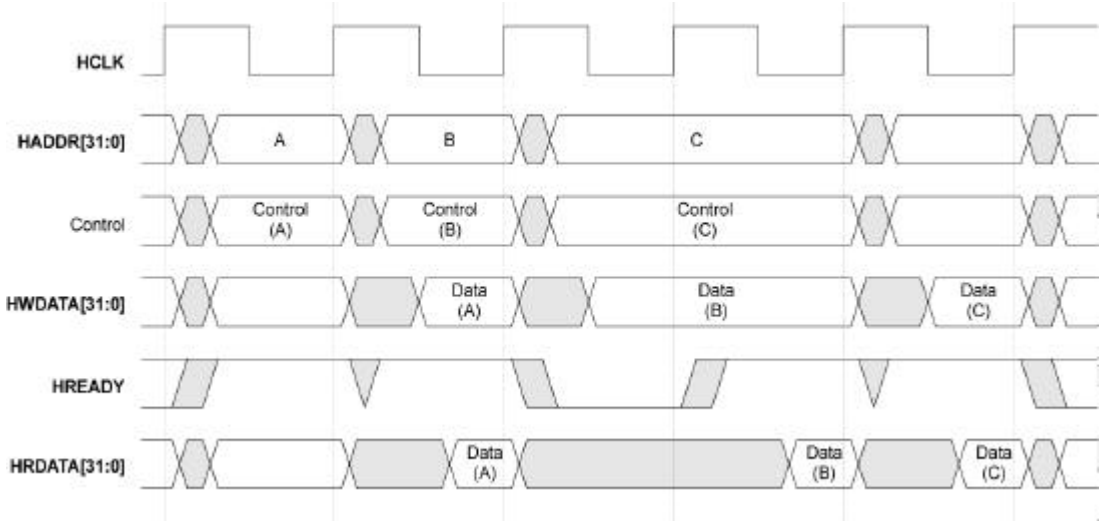


図 3-5 複数の転送

図 3-5 では、次のことが分かります。

- アドレス A と C への転送はともにウェイトステート無しです。
- アドレス B への転送はウェイトステート 1 回です。
- アドレス B への転送のデータフェーズを延長すると、その副作用として、アドレス C への転送のアドレスフェーズが延びます。

3.5 転送タイプ

あらゆる転送は、表 3-1 に示すように、**HTRANS[1:0]** 信号で示す 4 種類のタイプのうちの 1 つに分類できます。

表 3-1 転送タイプのエンコーディング

HTRANS[1:0]	タイプ	説明
00	IDLE	データ転送が要求されていないことを示します。バスマスターがバスに承認されているが、データ転送を実行したくないときは、IDLE 転送タイプが使用されます。 スレーブは IDLE 転送にウェイトステート無しの OKAY 応答を必ず返し、その転送はスレーブによって無視されます。
01	BUSY	BUSY 転送タイプの場合、バスマスターは転送バーストの途中で IDLE サイクルを挿入することができます。この転送タイプは、バスマスターが転送バーストを継続しているが、次の転送をすぐには実行できないことを示します。あるマスターが BUSY 転送タイプを使用すると、そのアドレスと制御信号がバーストの次の転送に反映される必要があります。 その転送はスレーブによって無視されます。スレーブは、IDLE 転送への応答と同様に、必ずウェイトステート無しの OKAY 応答を返します。
10	NONSEQ	バーストの最初の転送、またはシングル転送を示します。そのアドレスと制御信号は前回の転送とは無関係です。 バス上の複数のシングル転送は、複数のシングル転送バーストとして扱われます。したがって、その転送タイプは NONSEQUENTIAL です。
11	SEQ	バーストに残っている転送は SEQUENTIAL で、そのアドレスは前回の転送と関係があります。制御情報は前回の転送と同じです。アドレスは前回の転送のアドレスにそのサイズ (バイト単位) を加えたものと同じです。ラップ式バーストの場合、転送のアドレスは、転送内のビート数 (4、8、16 のいずれか) を乗算したサイズ (バイト単位) と等しいアドレス境界でラップします。

図 3-6 に、各転送タイプを示します。

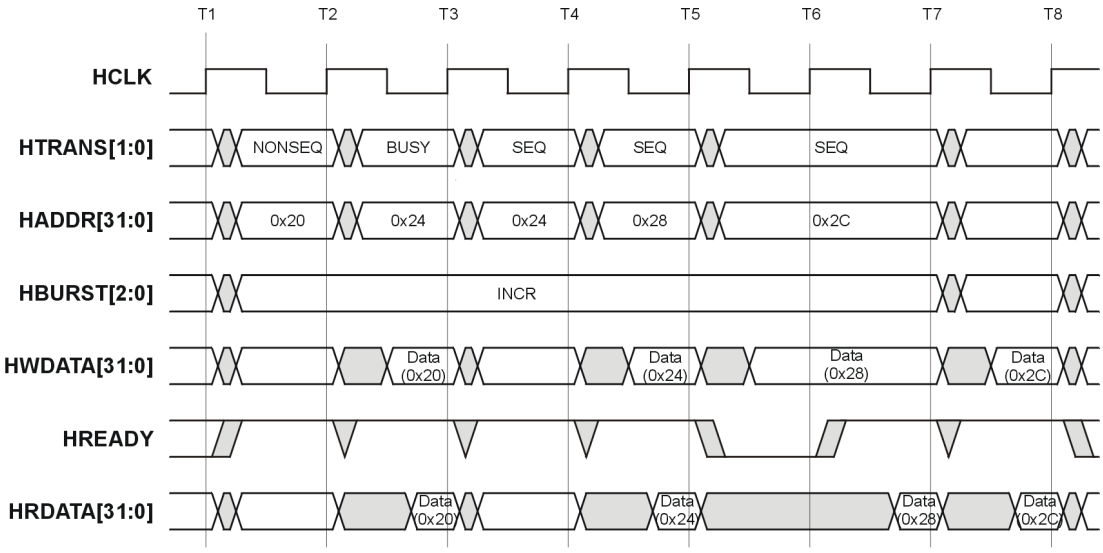


図 3-6 転送タイプの例

図 3-6 では、次のことが分かります。:

- 最初の転送は、バーストの始まりで、そのため NONSEQUENTIAL です。
- マスターは、そのバーストの 2 番目の転送をすぐには実行できないので、次の転送の開始を遅らせるために BUSY 転送を使用します。この例では、バースト内の次の転送を開始する準備ができるまでに、マスターは 1 サイクルだけ必要としています。つまり、ウェイトステート無しで完了します。
- マスターはすぐにバーストの 3 番目の転送を実行しますが、この時、スレーブは完了することができず、**HREADY** を使って 1 回のウェイトステートを挿入します。
- バーストの最後の転送は、ウェイトステート無しで完了します。

3.6 バースト動作

不定長バーストやシングル転送と同様に、4、8、および 16 ビートのバーストが AMBA AHB プロトコルに規定されています。インクリメント式とラップ式バーストの両者がそのプロトコルでサポートされています。

- インクリメント式バーストはシーケンシャルロケーションをアクセスします。そして、そのバーストの各転送のアドレスは前アドレスのインクリメント値となります。
- ラップ式バーストの場合、転送の開始アドレスはバースト内の総バイト数 (サイズ x ビート数) とは一致しない場合、そのバースト内の転送のアドレスは境界に到達するとラップします。例えば、ワード (4 バイト) アクセスの 4 ビートのラップ式バーストは 16 バイト境界でラップします。したがって、転送の開始アドレスが 0x34 の場合、バーストはアドレス 0x34、0x38、0x3C、0x30 への 4 つの転送を成立させます。

バースト情報は、**HBURST[2:0]** を使用して提供され、可能な 8 つのタイプが表 3-2 に定義されています。

表 3-2 バースト信号のエンコーディング

HBURST[2:0]	タイプ	説明
000	SINGLE	単独転送
001	INCR	不定長のインクリメント式バースト
010	WRAP4	4 ビートラップ式バースト
011	INCR4	4 ビート インクリメント式バースト
100	WRAP8	8 ビートラップ式バースト
101	INCR8	8 ビート インクリメント式バースト
110	WRAP16	16 ビートラップ式バースト
111	INCR16	16 ビート インクリメント式バースト

バーストは 1kB アドレス境界を越えてはいけません。したがって、この境界を超える固定長インクリメント式バーストをマスターが開始しないようにすることが重要です。

長さ 1 のバーストを持っている不定長インクリメント式バーストを使用してシングル転送を実行することは容認できます。

インクリメント式バーストは任意の長さをとることができますが、アドレスが 1kB 境界を超えてはいけないという制限があるため、上限が設けられています。

Note

バーストサイズは、転送されるバイト数ではなく、バーストのビート数を示します。1 回のバーストで転送されるデータの総数は、**HSIZE[2:0]** で示すように、各ビートのデータ数にビート数を乗算して計算します。

バースト内の転送はすべて、転送サイズと等しいアドレス境界に合わせる必要があります。例えば、ワード転送はワードアドレス境界 (すなわち、A[1:0] = 00) に、ハーフワード転送はハーフワードアドレス境界 (すなわち、A[0] = 0) に合わせる必要があります。

3.6.1 早期バースト終了

バーストが完了できない状況に陥ることがあります。したがって、バーストが早めに終了した場合に、バースト情報を使用するスレーブ設計が正しい動作手順をとれることが重要です。スレーブは、**HTRANS** 信号を監視し、そしてバースト開始後に、すべての転送に SEQUENTIAL または BUSY のラベルが付けられたかを確認することにより、バーストがいつ終了したかを判断できます。NONSEQUENTIAL または IDLE 転送が発生した場合、これは新しいバーストが開始し、前のバーストが終了していることを示します。

バスの所有権を失ったためにバスマスターがバーストを完了できなかった場合、バスマスターは今度のバスのアクセス権を得た時に、そのバーストを適切に再構築しなければなりません。例えば、4 ビートバーストのうちの 1 ビートを完了しただけの場合、マスターは不定長バーストを使用して残りの 3 つの転送を実行する必要があります。

以下のページに例を示します。

- 図 3-7 (P.3-13) に、*4 ビートラップ式バースト*を示します。
- 図 3-8 (P.3-14) に、*4 ビート インクリメント式バースト*を示します。
- 図 3-9 (P.3-15) に、*8 ビートラップ式バースト*を示します。
- 図 3-10 (P.3-15) に、*8 ビート インクリメント式バースト*を示します。
- 図 3-11 (P.3-16) に、*不定長バースト*を示します。

図 3-7 の例では、最初の転送にウェイトステートを追加した 4 ビートラップ式バーストを示します。

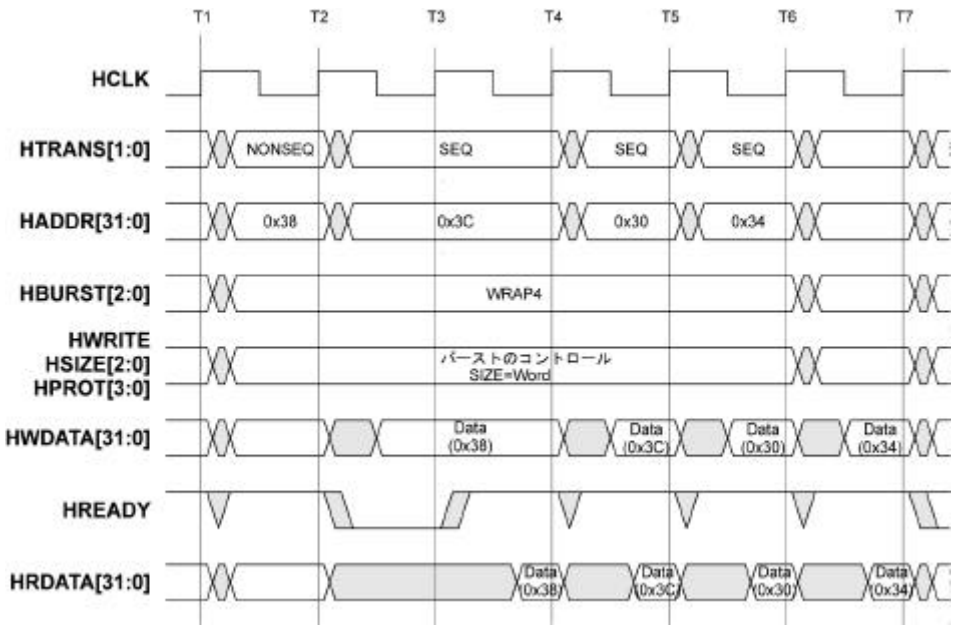


図 3-7 4 ビートラップ式バースト

このバーストはワード転送の 4 ビートバーストなので、そのアドレスは 16 バイト境界でラップし、その結果、アドレス 0x3C への転送の次にアドレス 0x30 への転送が続きます。インクリメント式バーストとの唯一の相違点は、P.3-14 の図 3-8 に示すように、アドレスが 16 バイト境界を越えて続くことです。

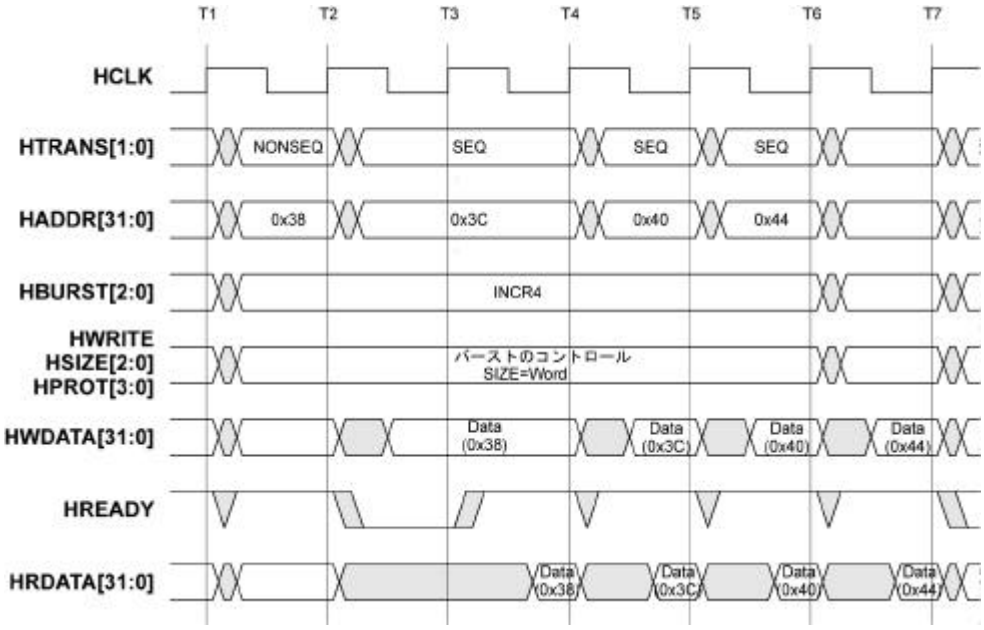


図 3-8 4 ビートインクリメント式バースト

図 3-9 の例では、ワード転送の 8 ビートバーストを示します。

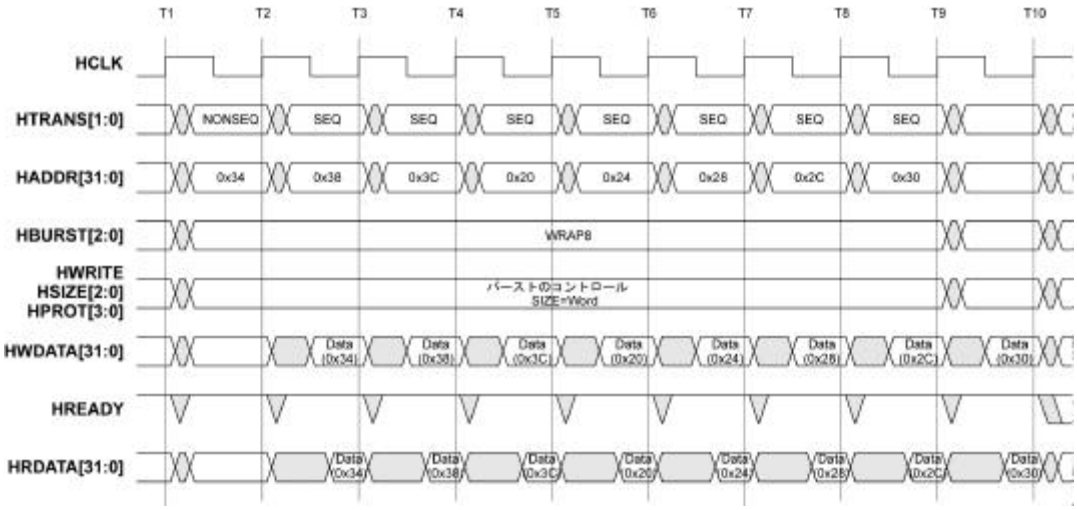


図 3-9 8 ビートラップ式バースト

アドレスは 32 バイト境界でラップし、そのため、アドレス 0x3C の次にアドレス 0x20 が続きます。

図 3-10 のバーストはハーフワード転送を使用するので、そのアドレスは 2 だけ増加します。このバーストはインクリメント式なので、そのアドレスは 16 バイト境界を越えて増加し続けます。

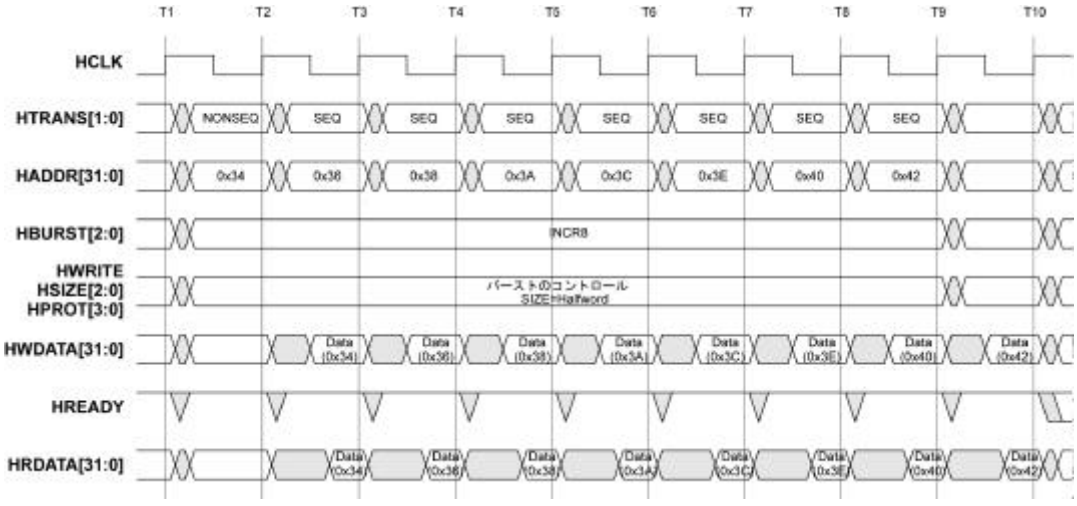


図 3-10 8 ビートインクリメント式バースト

図 3-11 の最後の例では、不定長のインクリメント式バーストを示します。

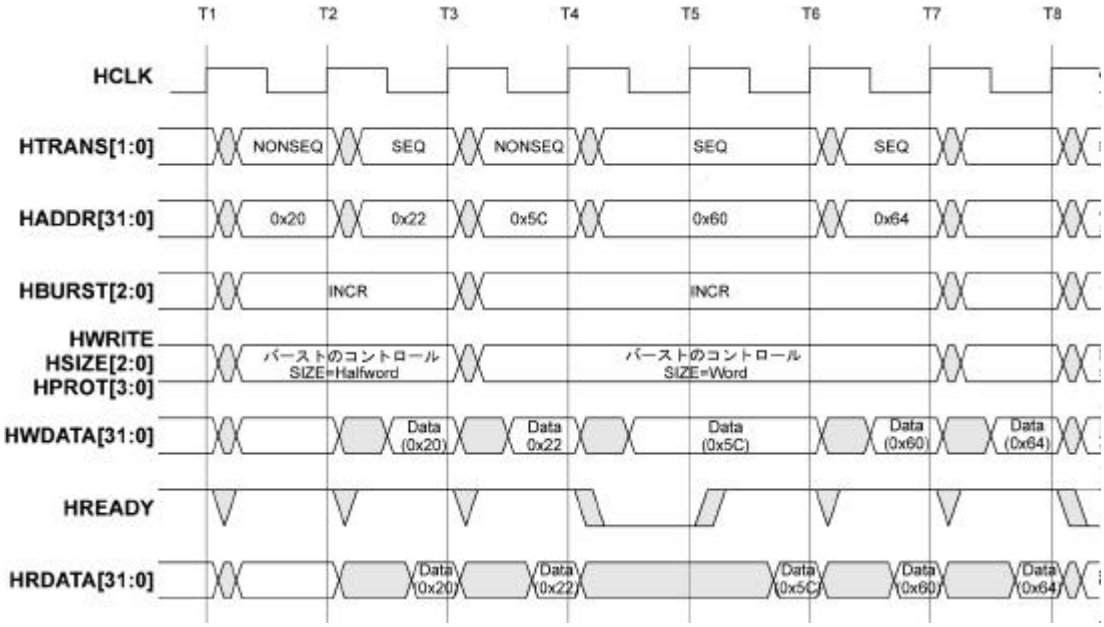


図 3-11 不定長バースト

図 3-11 では、以下の 2 つのバーストを示します。

- 2 つのハーフワード転送は、アドレス 0x20 から開始します。このハーフワード転送のアドレスは 2 だけ増加します。
- 3 つのワード転送は、アドレス 0x5C から開始します。このワード転送のアドレスは 4 だけ増加します。

3.7 制御信号

各転送は、転送タイプやバーストタイプだけでなく、転送に関する追加情報を提供する多くの制御信号を持っています。これらの制御信号のタイミングは、アドレスバスのタイミングとまったく同じです。しかし、転送バーストの間ずっと一定でなければなりません。

3.7.1 転送方向

HWRITE が HIGH のときは、この信号は書込み転送を示し、マスターはデータを書込みデータバス **HWDATA[31:0]** 上に送ります。LOW のときは、読出し転送が実行され、スレーブは読出しデータバス **HRDATA[31:0]** 上にデータを生成する必要があります。

3.7.2 転送サイズ

HSIZE[2:0] は、表 3-3 に示す転送サイズを知らせます。

表 3-3 サイズのエンコーディング

HSIZE[2]	HSIZE[1]	HSIZE[0]	サイズ	説明
0	0	0	8 ビット	バイト
0	0	1	16 ビット	ハーフワード
0	1	0	32 ビット	ワード
0	1	1	64 ビット	-
1	0	0	128 ビット	4 ワードライン
1	0	1	256 ビット	8 ワードライン
1	1	0	512 ビット	-
1	1	1	1024 ビット	-

サイズは、ラップ式バーストのアドレス境界を判断するために **HBURST[2:0]** 信号と共に使用します。

3.7.3 保護制御

保護制御信号 **HPROT[3:0]** はバスアクセスに関する追加情報を提供し、主に或るレベルのプロテクションを実装したいモジュールが使用するためのものです (表 3-4 参照)。

この信号は、以下の事項を示します。

- 転送がオペコードフェッチと、データアクセスのどちらか。
- 転送が特権モードアクセスと、ユーザーモードアクセスのどちらか。

メモリ管理装置付きのバスマスターの場合、この信号は現在のアクセスがキャッシュ可能かバッファ可能かも示します。

表 3-4 保護信号のエンコーディング

HPROT[3] キャッシュ可能	HPROT[2] バッファ可能	HPROT[1] 特権的	HPROT[0] データ / オペコード	説明
-	-	-	0	オペコードフェッチ
-	-	-	1	データアクセス
-	-	0	-	ユーザーアクセス
-	-	1	-	特権アクセス
-	0	-	-	バッファ不可
-	1	-	-	バッファ可能
0	-	-	-	キャッシュ不可
1	-	-	-	キャッシュ可能

すべてのバスマスターが正確なプロテクション情報を生成できるわけではありません。したがって、どうしても必要という場合でない限り、スレーブが **HPROT** 信号を使用しないようにすることを推奨します。

3.8 アドレスデコーディング

セントラルアドレスデコーダは、バス上の各スレーブの選択信号 **HSELx** を送出するために使用します。この選択信号は高位アドレス信号の組合せデコードです。複雑なデコードロジックを回避し、高速動作を保証するため、簡潔なアドレスデコーディング方式を提唱します。

HREADY が HIGH で、現在の転送が完了しつつあることを示している場合、スレーブはそのアドレスと制御信号および **HSELx** を取得することだけが求められます。ある状況で、**HREADY** が LOW のときに **HSELx** がアクティブになるが、現在の転送が完了する時までに、選択されたスレーブが変わってしまうということがあります。

1 つのスレーブに割り当て可能な最低アドレス空間は 1kB です。すべてのバスマスターは、1kB 境界を越えるインクリメント式転送を実行しないように設計され、その結果、バーストが決してアドレスデコード境界を越えないことを保証しています。

満杯に詰め込まれたメモリマップがシステム設計に組み込まれていない場合、存在していないアドレス位置へのアクセス時に応答を返すのに、追加のデフォルトスレーブを実装すべきです。存在しないアドレス位置への NONSEQUENTIAL または SEQUENTIAL 転送を試みた場合、そのデフォルトスレーブは ERROR 応答を返します。存在しない位置への IDLE または BUSY 転送の場合には、ウェイトステート無しの OKAY 応答を返します。通常、デフォルトスレーブの機能はセントラルアドレスデコーダの一部として実装されます。

図 3-12 には、代表的なアドレスデコーディングシステムとスレーブ選択信号を示します。

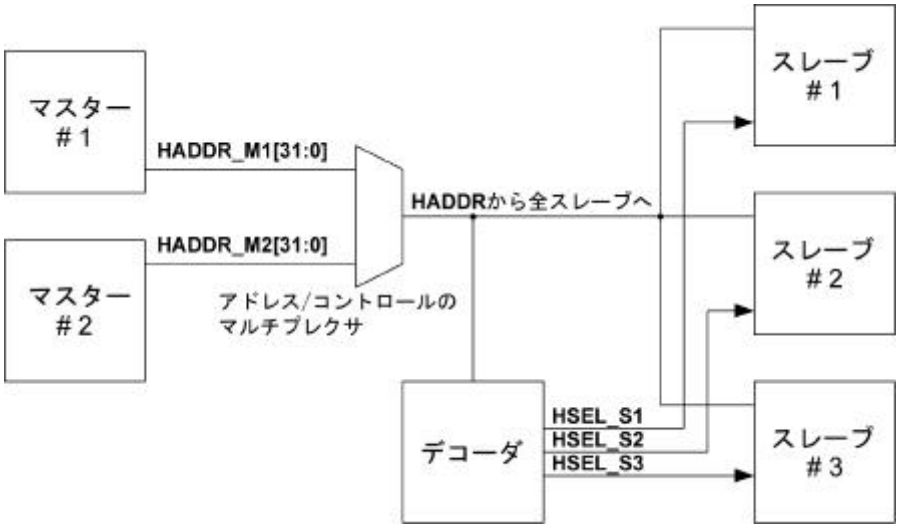


図 3-12 スレーブセレクト信号

3.9 スレーブ転送応答

マスターが転送を開始した後、スレーブはその転送の進捗具合を確認します。いったん開始した転送をバスマスターがキャンセルするための手段は、AHB 仕様書内には規定されていません。

スレーブは、アクセスされると必ず転送の状態を示す応答を返します。**HREADY** 信号は転送を延長するために使用され、これは転送の状態を示す応答信号 **HRESP[1:0]** と連携して機能します。

スレーブは、以下に示すように、多くの方法で転送を完了することができます。

- すぐに転送を完了します。
- 1 つ以上のウェイトステートを挿入し、転送を完了するための時間を確保します。
- エラー信号を送って、転送が失敗したことを知らせます。
- 転送の完了を遅らせるが、マスターとスレーブにバスを戻させ、そのバスが他の転送に利用できるようにします。

3.9.1 転送完了

HREADY 信号は、AHB 転送のデータ部分を延長するために使用します。LOW のとき、**HREADY** 信号は転送を延長する必要があることを示し、HIGH のときは、転送が完了できることを示します。

Note

バスアクセスの待ち時間を計算できるようにするため、すべてのスレーブは、バスを戻す前に挿入する定められた最大数のウェイトステートを持っている必要があります。多数のクロックサイクルの間、1 つのアクセスがバスをロックしないように、スレーブが 16 を超えるウェイトステートを挿入しないようにすることを推奨します。しかし、これは必須ではありません。

3.9.2 転送応答

代表的なスレーブは **HREADY** 信号を使用して、適切な数のウェイトステートを転送に挿入します。その結果、その転送は **HREADY** HIGH 状態と OKAY 応答により完了します。これは転送が成功した上で完了したことを示します。

ERROR 応答は、スレーブが使用して、関連転送におけるエラー状態を示します。通常、これは読出し専用メモリ位置への書込みなどのプロテクションエラーに対して使用します。

SPLIT 応答と RETRY 応答を組み合わせた信号は、スレーブに転送の完了を遅らせますが、他のマスターが使用できるようにバスを解放します。これらの組合せは通常、アクセスの待ち時間が長く、かつこれらの応答コードを利用できるスレーブだけが必要とし、他のマスターが長時間バスにアクセスしないようにするために使用します。

SPLIT と RETRY 動作の詳細については、P.3-24「*分割および再試行*」を参照して下さい。

HRESP[1:0] のエンコーディング、転送応答信号、各応答の説明を、表 3-5 に示します。

表 3-5 応答のエンコーディング

HRESP[1]	HRESP[0]	応答	説明
0	0	OKAY	**HREADY** が HIGH のとき、これは転送が成功した上で完了したことを示します。OKAY 応答は、他の 3 つの応答のうちのどれかを出す前に、**HREADY** LOW 状態で挿入される追加サイクルに対しても使用します。
0	1	ERROR	この応答はエラーが発生したことを示します。転送が失敗したことに気付くように、このエラー状態をバスマスターに信号で知らせるべきです。 エラー状態には 2 サイクルの応答が必要です。
1	0	RETRY	RETRY 応答は、転送がまだ完了していないことを示します。その結果、バスマスターは転送の再試行を行ないます。マスターは、転送が完了するまで、再試行を繰り返します。 2 サイクルの RETRY 応答が必要です。
1	1	SPLIT	転送がまだ成功した上で完了していません。バスへのアクセスが次に許可されたとき、バスマスターは転送を再試行します。スレーブは、転送が完了できるとき、マスターの代わりにバスへのアクセスを要求します。 2 サイクルの SPLIT 応答が必要です。

どの応答が発行されるか判断する前にスレーブが多くウェイトステートを挿入する必要がある場合、応答を OKAY にする必要があります。

3.9.3 2 サイクル応答

OKAY 応答だけが 1 サイクルで出すことができます。ERROR、SPLIT、および RETRY 応答は 2 サイクル以上必要です。これらの応答のいずれかで完了するためには、スレーブは、最後から 2 番目のサイクルにおいて **HRESP[1:0]** を駆動して ERROR、RETRY、または SPLIT を示し、その一方で、**HREADY** を LOW にして追加サイクルの転送を延長します。最終サイクルでは、**HREADY** が HIGH になり転送を終了し、その一方で、**HRESP[1:0]** は駆動されたままの状態に保持され、ERROR、RETRY、または SPLIT を示します。

ERROR、SPLIT、または RETRY 応答を出すために、スレーブが 3 サイクル以上必要となる場合、転送の開始時に追加のウェイトステートを挿入してもかまいません。この時、**HREADY** 信号は LOW になり、応答は OKAY に設定しなければなりません。

2 サイクル応答はバスのパイプライン処理性のために必要です。スレーブが ERROR、SPLIT、RETRY 応答のいずれかを発行し始める時までに、次の転送のアドレスは既にバス上に送信されてあります。この 2 サイクルの応答により、次の転送の開始前に、マスターがこのアドレスをキャンセルし、**HTRANS[1:0]** を IDLE する十分な時間が得られます。

SPLIT と RETRY 応答の場合、次の転送は現在の転送が完了する前に実行してはいけないので、キャンセルする必要があります。しかし、ERROR 応答については、現在の転送が繰り返されない場合、次の転送は完了してもしなくてもかまいません。

図 3-13 では、RETRY 動作の例を示します。

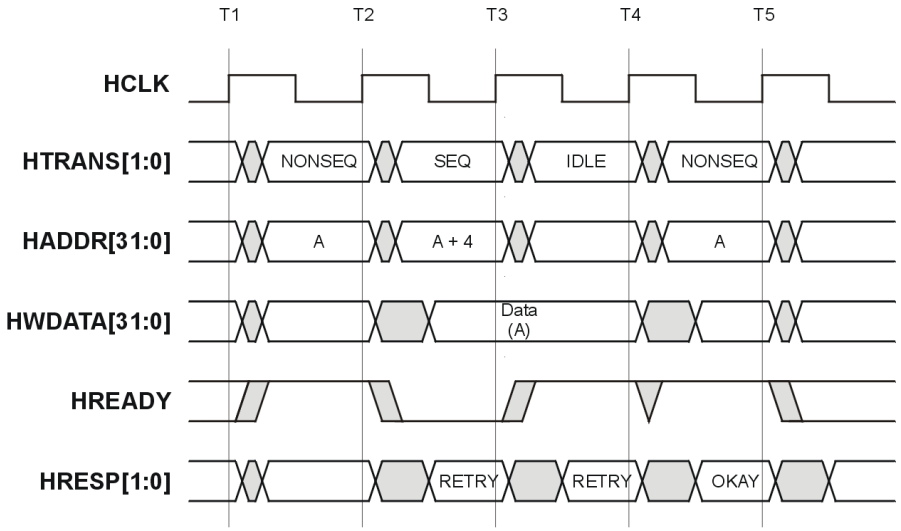


図 3-13 再試行応答での転送

以下のイベントについて説明します。

- マスターがアドレス A への転送を開始します。
- この転送についての応答を受け取る前に、マスターがアドレスを A + 4 に進めます。
- アドレス A のスレーブがすぐには転送を完了できず、そのため RETRY 応答を発行します。この応答はマスターに、アドレス A の転送が完了できず、そのためアドレス A + 4 の転送がキャンセルされ、IDLE 転送に置き換えられることを知らせます。

図 3-14 では、(**HRESP** が OKAY を示す時に) 発行予定の応答について判断するためにスレーブが 1 サイクル必要とし、そのスレーブが 2 サイクルの ERROR 応答で転送を終了する場合の転送を示します。

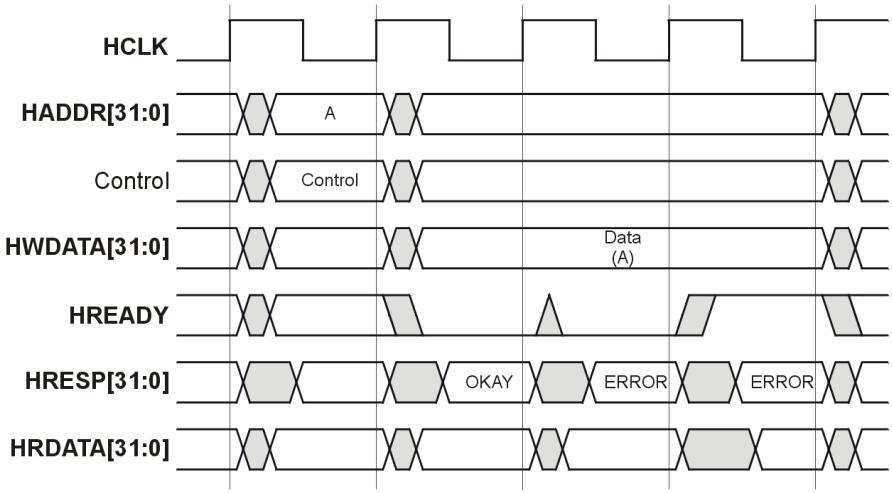


図 3-14 エラー応答

3.9.4 エラーー応答

スレーブが ERROR 応答を発行すると、マスターはバースト内の残りの転送をキャンセルする場合があります。しかし、これは厳密な必要条件ではなく、マスターがバースト内の残りの転送を続行しても容認できます。

3.9.5 分割および再試行

SPLIT と RETRY 応答により、スレーブが転送データをすぐに供給できないときにバスを解放するメカニズムが得られます。また、両方のメカニズムにより、その転送がバス上で終了でき、その結果、優先順位がより高いマスターがバスへのアクセス権を得ることができます。

SPLIT と RETRY 間の相違は、SPLIT または RETRY が発生した後にアービタがバスを割り当てる方法です。

- RETRY の場合、アービタは通常優先順位方式を使用し続け、その結果、より高い優先順位を持っているマスターだけがバスへのアクセス権を取得します。
- SPLIT 転送の場合、アービタは、たとえ優先順位が低くてもバスを要求した他のマスターがアクセス権を得られるように、優先順位方式を調整します。SPLIT 転送が完了するためには、スレーブがデータを持っているときに、アービタに通知しなければいけません。

SPLIT 転送では、スレーブとアービタの両者において複雑さが増大しますが、他のマスターが使用できるようにバスを完全に解放するという利点があります。ところが、RETRY の場合、優先順位がより高いマスターだけがバスにアクセスできます。

バスマスターは SPLIT と RETRY を同じように取り扱います。バスマスターは、転送が成功した上で完了するか、ERROR 応答で終了するまでバスを要求し、転送を試み続けます。

3.10 データバス

トライステートドライバを使用しないで AHB システムの実装を可能にするためには、独立した読出しデータバスと書込みデータバスが必要です。最小データバス幅は 32 ビットと規定されていますが、P.3-41 「*AHB データバス幅について*」で説明するように、その幅は増加できます。

3.10.1 HWDATA[31:0]

書込みデータバスは、書込み転送中、バスマスターによって駆動されます。転送を延長すると、バスマスターは、転送が完了し、**HREADY** HIGH になるまで、データを有効に保持します。

転送はすべて、転送サイズと等しいアドレス境界に合わせる必要があります。例えば、ワード転送はワードアドレス境界 (すなわち、A[1:0] = 00) に、ハーフワード転送はハーフワードアドレス境界 (すなわち、A[0] = 0) に合わせる必要があります。

32 ビットバスでの 16 ビット転送など、バス幅より狭い転送の場合、バスマスターは該当するバイトレーンを駆動するだけですみます。スレーブは正しいバイトレーンの中から書込みデータを選択します。P.3-26 の表 3-6 と表 3-7 では、リトルエンディアンシステムとビッグエンディアンシステムのそれぞれに対してどのバイトレーンがアクティブかを示します。この情報は、必要に応じて、より広いデータバスの実装にも拡張できます。データバスの幅より小さい転送サイズを持ったバースト転送は、バーストのビートごとに異なったアクティブバイトレーンを持っています。

アクティブバイトレーンはシステムのエンディアン方法に依存するが、AHB は必要なエンディアン方法を指定しません。したがって、バス上の全マスターと全スレーブが同じエンディアン方法を持っていることが重要です。

3.10.2 HRDATA[31:0]

読出しデータバスは、読出し転送中、当該スレーブによって駆動されます。スレーブが **HREADY** を LOW に保持して読出し転送を延長した場合、スレーブは **HREADY** HIGH レベルを示し、転送の最終サイクルの終わりで有効データを供給する必要があるだけです。

バス幅より狭い転送の場合、表 3-6 と表 3-7 に示すように、スレーブはアクティブバイトレーン上に有効データを供給することが求められるだけです。バスマスターは、正しいバイトレーンの中からデータを選択する責任があります。

転送が OKAY 応答で完了したとき、スレーブは有効データを供給するだけで済みます。SPLIT、RETRY、ERROR 応答の場合、有効読出しデータは必要ありません。

表 3-6 32 ビットリトルエンディアンデータバスのアクティブバイトレーン

転送サイズ	アドレスオフセット	データ [31:24]	データ [23:16]	データ [15:8]	データ [7:0]
ワード	0	✓	✓	✓	✓
ハーフワード	0	-	-	✓	✓
ハーフワード	2	✓	✓	-	-
バイト	0	-	-	-	✓
バイト	1	-	-	✓	-
バイト	2	-	✓	-	-
バイト	3	✓	-	-	-

表 3-7 32 ビットビッグエンディアンデータバスのアクティブバイトレーン

転送サイズ	アドレスオフセット	データ [31:24]	データ [23:16]	データ [15:8]	データ [7:0]
ワード	0	✓	✓	✓	✓
ハーフワード	0	✓	✓	-	-
ハーフワード	2	-	-	✓	✓
バイト	0	✓	-	-	-
バイト	1	-	✓	-	-
バイト	2	-	-	✓	-
バイト	3	-	-	-	✓

3.10.3 エンディアン方法

システムが正しく機能するためには、全モジュールが同じエンディアン方法を持ち、かつデータ経路やブリッジも同じエンディアン方法を持っていることが必須です。

組込みシステムのほとんどでは、幅広いシリコンの重大なオーバーヘッドを引き起こすという理由で、変動するエンディアン方法をサポートしていません。

モジュール設計者には、エンディアン方法を選択するための設定ピンか、内部制御ビットのどちらかを使用して、広範囲の用途に使用されるモジュールだけを 2 種のエンディアン設定することを推奨します。多くの特定用途向けブロックに対しては、そのエンディアン方法をリトルエンディアンか、ビッグエンディアンのどちらかに固定すると、インターフェースを小型化、低電力化、高性能化することができます。

3.11 アービトレーション

アービトレーションメカニズムは、一度に 1 つだけのマスターがバスにアクセスできるようにするために使用します。アービタは、多くのバス使用の要求を注視し、そしてバスの使用を要求しているマスターの中でどのマスターの優先順位が現在最も高いかを判断することにより、この機能を果たしています。また、アービタは SPLIT 転送を完了したがっているスレーブからの要求も受け取ります。

SPLIT 転送を行えないスレーブは、アービトレーションプロセスを意識する必要がありません。ただし、バスの所有権が移ると、転送のバーストが完了しない場合もあるという事に気をつける必要があります。

3.11.1 信号の説明

各アービトレーション信号について以下に簡単に説明します。

HBUSREQx このバス要求信号は、バスへのアクセス権を要求するためにバス マスターが使用します。各バスマスターは個々に、アービタへの **HBUSREQx** 信号を持っており、どのシステムでも最大 16 の独立したバスマスターが存在可能です。

HLOCKx ロック信号は、マスターがバス要求信号と同時にアクティブにします。これはマスターが多数の分割不可能な転送を実行していることをアービタに知らせます。アービタは、ロックされた転送の最初の転送が始まると、他のバスマスターにバスへのアクセス権を与えません。アービタが認可信号を変更するのを防ぐため、アービタが参照するアドレスの 1 サイクル以上前に **HLOCKx** をアクティブにする必要があります。

HGRANTx 認可信号は、アービタが生成し、ロックされた転送と SPLIT 転送を考慮に入れて、バスを要求しているマスターの中で使用中のマスターが現在優先順位が最も高いことを示します。
HCLK の立上がりエッジ時に、**HGRANTx** が HIGH、**HREADY** が HIGH のとき、マスターがアドレスバスの所有権を取得します。

HMASTER[3:0] アービタは、**HMASTER[3:0]** 信号を使用して、現在どのマスターにバスの使用が認可されているかを示します。これはセントラルアドレス・制御マルチプレクサを制御するために使用できます。SPLIT トランザクションを完了できるのはどのマスターかをアービタに知らせることができるように、マスター番号も SPLIT 可能スレーブによって要求されます。

HMASTLOCK アービタは、**HMASTLOCK** 信号をアクティブにして、現在の転送がロックされた転送シーケンスの一部であることを知らせます。この信号のタイミングはアドレスと制御信号のタイミングと同じです。

HSPLIT[15:0] 16 ビット*分割完了*バスは、SPLIT 可能スレーブが使用して、どのバスマスターが SPLIT トランザクションを完了できるかを示します。この情報はアービタが必要とし、これに基づいて転送を完了するため、バスへのアクセス権がマスターに与えられます。

詳細については、以下を参照して下さい。
- *バスアクセスの要求*
- *バスアクセスの認可*：P.3-30
- *早期バースト終了*：P.3-33
- *ロックされた転送*：P.3-34

3.11.2 バスアクセスの要求

バスマスターは **HBUSREQx** 信号を使用して、バスへのアクセス権を要求します。任意のサイクル中にバスを要求できます。アービタはクロックの立上がり時に要求を取得し、内部の優先順位アルゴリズムを使用して、バスへのアクセス権を次に取得するのはどのマスターなのかを決めます。

アービタは通常、バーストが完了しつつあるとき、別のバスマスターを認可するだけです。しかし必要に応じて、より高い優先順位のマスターにバスへのアクセス権を与えるために、アービタはバーストを早めに終了することが出来ます。

マスターは、ロックされたアクセスを要求すると、他のマスターにバスの使用を認可しないようにアービタに通知するために、**HLOCKx** 信号をアクティブにしなければなりません。

マスターにバスの使用を認可し、固定長バーストを実行しているときは、そのバーストを完了するためにバスを要求し続ける必要はありません。アービタはバーストの進捗状況を観察し、そして **HBURST[2:0]** 信号を使用して、マスターが要求している転送数を判断します。マスターが現在進行しているバーストの後に別のバーストを実行したい場合、そのバースト時に要求信号を再度アクティブにすべきです。

マスターがバーストの途中でバスへのアクセス権を失った場合、バスへのアクセス権を再取得するために、**HBUSREQx** の要求ラインを再度アクティブにしなければなりません。

不定長バーストの場合、マスターは、最後の転送が開始されるまで、その要求をアクティブにし続けるべきです。アービタは、不定長バーストの終了時点では、そのアービトレーションをいつ変更すべきか予測できません。

マスターは、バスの使用を要求していないのにバスの使用が認可されることがあります。これは、どのマスターもバスを要求していなく、アービタがデフォルトマスターにアクセス権を認可している場合に起こります。したがって、マスターがバスへのアクセス権を必要としていない場合、転送タイプを **HTRANS** にして、IDLE 転送を示すことが重要です。

3.11.3 バスアクセスの認可

アービタは、適切な **HGRANTx** 信号をアクティブにすることにより、バスを要求しているマスターの中で優先順位が現在最も高いのはどれかを示します。**HREADY** HIGH レベルを示し、現在の転送が完了したとき、そのマスターは認可され、アービタは **HMASTER[3:0]** 信号を変更してそのバスマスター番号を示します。

図 3-15 では、全転送がウェイトステートが無い場合で、**HREADY** 信号が HIGH になるときのプロセスを示します。

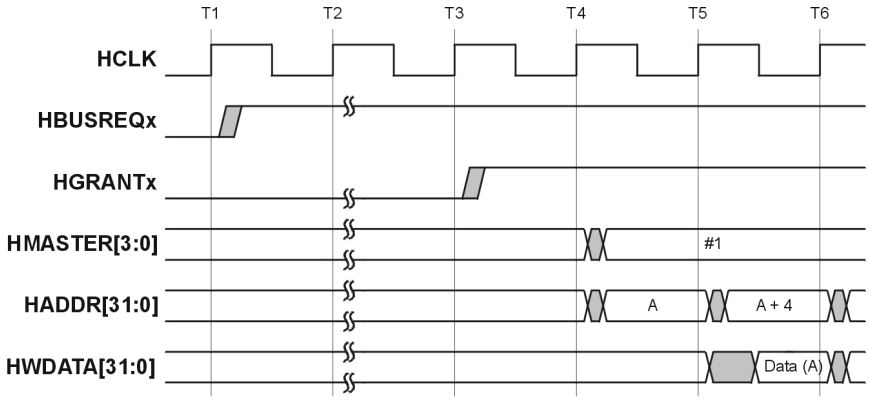


図 3-15 ウェイトステートがないアクセスの認可

図 3-16 では、ウェイトステートがバスハンドオーバーに与える影響を示します。

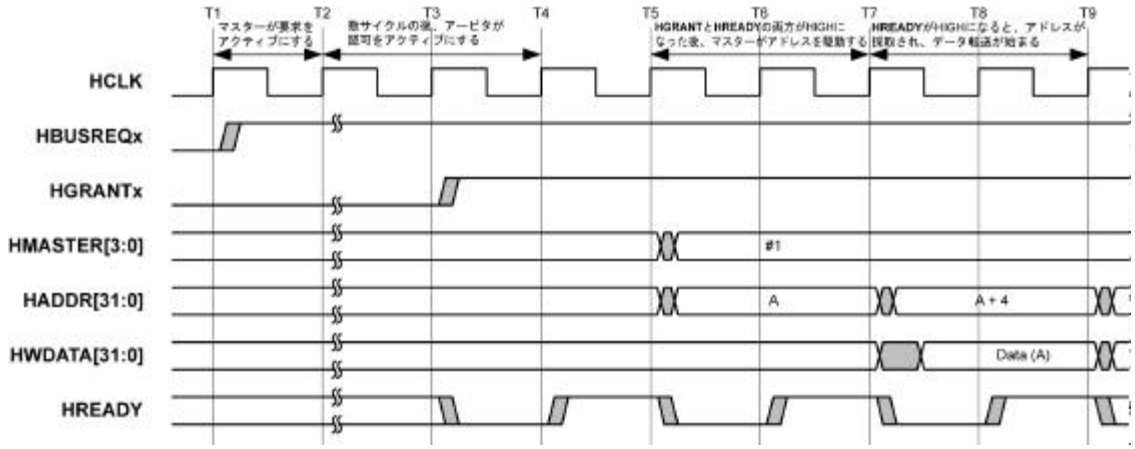


図 3-16 ウェイトステートがあるアクセスの認可

データバスの所有権はアドレスバスの所有権より遅れます。**HREADY** は HIGH レベルを示し、転送が完了すると、アドレスバスを所有しているマスターはデータバスを使用できるようになり、その転送が完了するまでデータバスを所有し続けます。図 3-17 では、2 つのバスマスター間でハンドオーバーが発生したときに、データバスの所有権がどのように移動するかを示します。

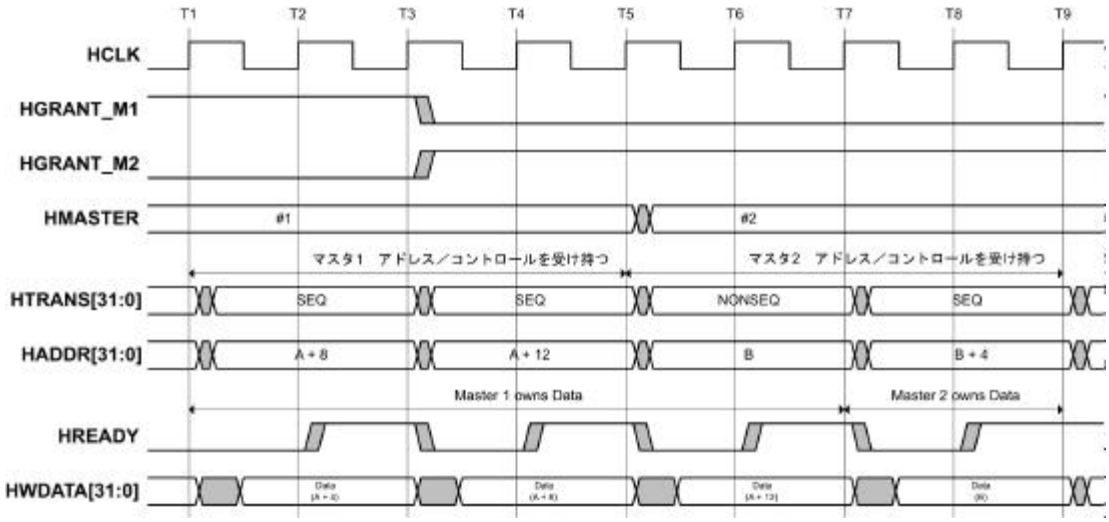


図 3-17 データバスの所有権

図 3-18 では、転送バーストの終了時に、アービタがどのようにしてバスを渡すことができるかを例を挙げて示します。

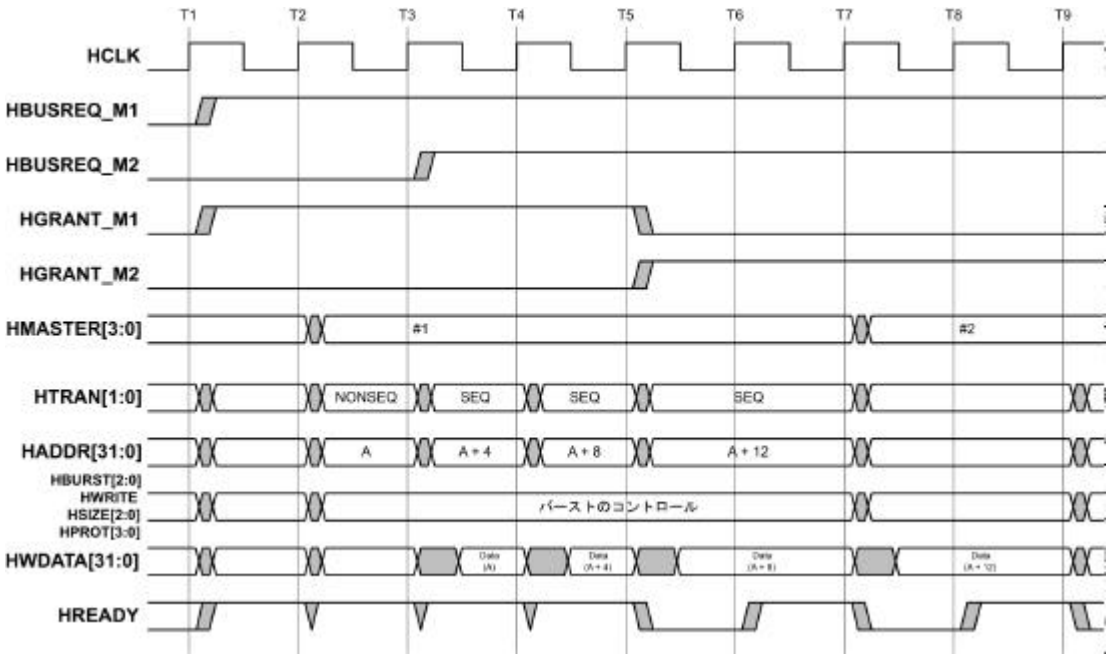


図 3-18 バースト後のハンドオーバー

最後から 2 番目のアドレスが取得されると、アービタは HGRANTx 信号を変更します。その新しい HGRANTx 情報は、バーストの最後のアドレスを取得するポイントと同じポイントで取得されます。

図 3-19 では、**HGRANTx** と **HMASTER** 信号がシステムでどのように使用されるかを示します。

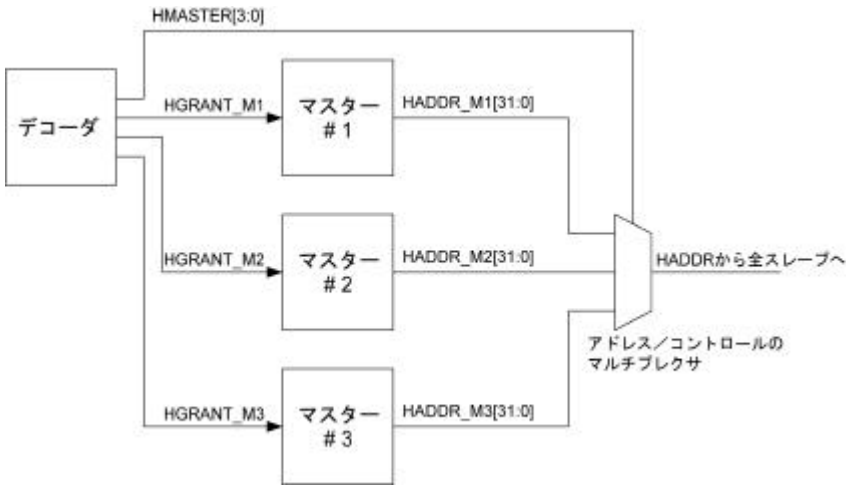


図 3-19 バスマスター認可信号

Note

セントラルマルチプレクサが使用されているので、各マスターはすぐに実行したい転送のアドレスを送出でき、バスの使用が認可されるまで待つ必要はありません。HGRANTx 信号は、マスターだけが使用して、いつバスを所有するか、つまり、いつの時点でアドレスが適切なスレーブによって取得されたかを判断します。

HMASTER バスの遅延版は、書込みデータマルチプレクサを制御するために使用します。

3.11.4 早期バースト終了

アービタは通常、転送バーストが終了するまで、新しいマスターにバスを渡すことはしません。しかし、バスへの過剰なアクセス時間を防止するために、バーストを早めに終了させる必要があるとアービタが判断した場合、バーストが完了する前に、アービタは他のバスマスターに認可を渡す場合があります。

バーストの途中でバスの所有権を失った場合、マスターはそのバーストを完了するために、バスについてアービトレーションを再度行なう必要があります。マスターは、完全な 4、8、および 16 ビートバーストをもはや実行する必要はないという事実を反映するように HBURST と HTRANS 信号を合わせる必要があります。

例えば、マスターが 8 ビートバーストでの 3 回の転送を完了することができる場合、マスターは、バスへのアクセス権を再取得した時に正当なバーストエンコーディングを使用して残りの 5 つの転送を完了する必要があります。任意の正当な組合せが使用できるので、5 ビート不定長バーストか、または 1 ビート不定長バーストを伴う 4 ビート固定長バーストのどちらか一方は容認できます。

3.11.5 ロックされた転送

アービタは各マスターからの **HLOCKx** 信号を観察し、マスターがロックされた転送シーケンスをいつ実行したがっているかを判断します。そのため、アービタはそのロックされた転送シーケンスが完了するまで他のバスマスターにバスへのアクセス権を認可しないことを保証する責任があります。

ロックされた転送シーケンスの後、アービタは追加の転送に備えてバスマスターを常に認可状態に保持して、そのロックされた転送シーケンスの最後の転送が成功した上で完了し、SPLIT や RETRY 応答を受け取らないことを保証します。したがって、ロックされた転送シーケンスの後にマスターが IDLE 転送を挿入し、これにより別の転送バーストを開始する前にアービトレーションが変わる機会を提供することを推奨します。しかし、これは必須ではありません。

アービタには、**HMASTLOCK** 信号をアクティブにする責任もあります。この信号のタイミングはアドレスと制御信号のタイミングと同じです。この信号は、現在の転送がロックされており、したがって、バスへのアクセス権が他のマスターに認可される前に処理しなければならないことを全スレーブに知らせます。

3.11.6 デフォルトバスマスター

あらゆるシステムは、他の全てのマスターがバスを使用できない場合に、バスへのアクセス権が認可されるデフォルトバスマスターを組み込む必要があります。認可されるとデフォルトバスマスターはただ、IDLE 転送を行なうだけです。

どのマスターもバスを要求していない場合、アービタはデフォルトマスターに認可を与えるか、またはバスアクセスの遅延が短いということによる利点を持ったマスターに認可を与えることになります。

バスに対して、デフォルトマスターにアクセス権が認可されると、新しい転送がバス上で開始される事が無いようにする効果的な機構が動きます。さらに、それは低電力動作モードに入る前に実行すべき効果的なステップになります。

他のマスターが完了すべき SPLIT 転送を待っている場合、デフォルトマスターに認可を与えます。

3.12 分割転送

SPLIT 転送は、アドレスをスレーブに与えているマスターの動作を、適切なデータを使って応答するスレーブの動作から分離（または分割）することによってバスの総合利用率を向上させます。

転送が生じたとき、スレーブは、その転送の実行に非常に多数のサイクルが必要と判断し、SPLIT 応答を発行する決定を下せます。この応答は、スレーブが転送を完了する準備ができたことを知らせるまで、転送を試みているマスターにはバスへのアクセス権を認可すべきではないと、アービタに知らせます。したがって、アービタには、応答信号を観察し、内部的には分割されたマスターからの要求をマスクする責任があります。

転送のアドレスフェーズの間、アービタはその転送を行なっているマスターを識別する **HMASTER[3:0]** 上に、タグすなわちマスター番号を生成します。SPLIT 応答を発行するスレーブは、転送を完了できること知らせることができなければなりません。スレーブはこれを、**HMASTER[3:0]** 信号上のマスター番号を記録することにより行ないます。

その後で、スレーブは転送を完了できるとき、スレーブからアービタへの **HSPLITx[15:0]** 信号上のマスター番号にしたがって適切なビットをアクティブにします。アービタはその時、マスターから要求されたマスクを取るのに、この情報を使用します。やがて、マスターは転送を再試行するのにアクセス権を認可することになります。アービタはサイクルごとに **HSPLITx** バスを取得します。したがって、スレーブは、アービタにそれを再認識させる目的で、1 サイクルの間適切なビットをアクティブにすることのみ必要とします。

複数の SPLIT 可能スレーブが存在するシステムでは、各スレーブからの **HSPLITx** バスの論理和をとり、1 つの結果としての **HSPLIT** バスをアービタに提供することができます。

ほとんどのシステムでは、16 台という最大数のバスマスターを使用することはありません。したがって、アービタは、バスマスターと同じ数のビットを持つ単一の **HSPLIT** バスを必要とするだけです。しかし、最大 16 台のマスターをサポート出来るように全 SPLIT 可能スレーブを設計することを推奨します。

3.12.1 分割転送手順

SPLIT トランザクションの基本的段階は以下のとおりです。

1. マスターは、他の転送とまったく同じ方法で転送を開始し、アドレスと制御情報を発行します。
2. スレーブがすぐにデータを供給できる場合は、そうしてもかまいません。そのデータを取得するのに多数のサイクルが必要とスレーブが判断した場合は、スレーブが SPLIT 転送応答を発行します。

   各転送中に、アービタが番号、またはタグを送信し、どのマスターがバスを使用しているかを示します。スレーブは後程の転送再開に役立てるために、この番号を記録します。
3. アービタは、他のマスターにバスの使用を認可し、SPLIT 応答の動作がバスマスターハンドオーバーを発生させます。他のすべてのマスターも SPLIT 応答を受け取ると、デフォルトマスターが認可されます。
4. 転送を完了する準備ができているとき、スレーブはアービタへの **HSPLITx** バスの該当ビットをアクティブにして、どのマスターにバスへのアクセス権を認可すべきかを示します。
5. アービタは、各サイクル時に、**HSPLITx** 信号を観察し、**HSPLITx** の任意のビットがアクティブになったとき、適切なマスターの優先順位を復元します。
6. 結局、アービタがそのマスターに認可を与え、その結果、そのマスターは転送を再度試みることができます。優先順位がより高いマスターがバスを使用している場合、すぐに再試行しないことがあります。
7. 転送が最終的に発生したとき、スレーブは OKAY 転送応答で終了します。

3.12.2 複数の分割転送

バスプロトコルは、バスマスター 1 台当たり 1 つの未処理トランザクションを許可します。マスターモジュールが 2 つ以上の未処理トランザクションを処理できる場合、各未処理トランザクション用の要求と認可信号のセットを追加する必要があります。プロトコルレベルでは、1 つのモジュールは多くの異なったバスマスターとして機能し、その各々が 1 つの未処理トランザクションを持つことできます。

しかし、SPLIT 可能スレーブは同時に処理できる数以上の転送要求を受け取ることがあります。これが発生した場合、スレーブがその転送についてのアドレスと制御情報を記録せずに、SPLIT 応答を発行することが許容されます。そしてスレーブにはバスマスター番号を記録することだけが求められます。スレーブは、前に分割したが、アドレスと制御情報を記録しなかった全マスターのために **HSPLITx** バス上の該当ビットをアクティブにすることによって、別の転送を処理可能であることを示すことができます。

そして、アービタはそれらのマスターにバスへのアクセス権を再度認可することができ、それらのマスターは転送を再試行し、スレーブが要求しているアドレスと制御情報を返します。これは、マスターが要求している転送を完了することが許可されるまで、マスターには何度でもバスへのアクセス権を認可できることを意味します。

3.12.3 デッドロックの防止

SPLIT と RETRY 転送応答は両者とも、バスのデッドロックが発生しないように注意して使用する必要があります。各スレーブは所定のサイクル数以内で転送を終了するように設計されているので、単独の転送が AHB をロックする可能性はありません。しかし、SPLIT または RETRY 応答を発行するスレーブに、多くのマスターがそのスレーブが処理できない方法でアクセスしようとした場合、デッドロックが発生することはあり得ます。

分割転送

SPLIT 転送応答を発行できるスレーブの場合、システム内の最大で 16 台のマスターのそれぞれから発行される要求に対してスレーブが確実に対応できるようにして、デッドロックを防止します。スレーブは、各転送についてのアドレスと制御情報を保管する必要はなく、ただ単に、転送要求が行なわれ、SPLIT 応答が発行されたという事実を記録します。最終的に、全マスターは低い優先順位になり、それによりスレーブは要求にしたがって順序正しく作業し、どの要求を処理しているかをアービタに知らせます。このようにスレーブは全要求が最終的に処理されることを保証します。

スレーブは、多くの未処理の要求を持っているとき、ロックされた転送を他の転送より先に完了しなければならないことを知っていながら、その未処理の要求を任意の順番で処理しようとする場合があります。

スレーブがアドレスと制御情報をラッチしないで、SPLIT 応答を使用することは正当です。スレーブはその特定のマスターが転送を試みたことを記録するだけで、その後ある時点で、転送を完了する準備ができたという通知によりアドレスと制御情報を取得できます。そのマスターはバスの使用が認可され、転送を再開します。これにより、スレーブはアドレスと制御情報をラッチし、すぐにそのデータに応答するか、または多数のサイクル数の追加が必要な場合に、別の SPLIT 応答を発行することができます。

理想的には、スレーブは対応できる数以上の未処理の転送を持つべきではありませんが、これに対応するためのメカニズムはバスのデッドロックを防止するために必要です。

再試行転送

RETRY 応答を発行するスレーブには、一度に 1 つのマスターだけがアクセスできます。これはバスのプロトコルによっては実現していないので、システムアーキテクチャによって保証すべきです。ほとんどの場合、RETRY 応答を発行するスレーブは、一度に 1 つのマスターだけがアクセスできる周辺ユニットです。したがって、これはより高いレベルのプロトコルによって保証されます。

RETRY スレーブにアクセスする複数のマスターに対するハードウェア保護は、そのプロトコルの必要条件ではなく、以下のパラグラフで説明するように実装してもかまいません。バスレベルの唯一の必要条件は、スレーブは所定のクロックサイクル数以内で **HREADY** を HIGH にしなければならないということです。

ハードウェア保護が必要な場合、RETRY スレーブ自体の内部に実装してもかまいません。スレーブは、RETRY を発行した時、マスター番号を取得できます。その時点と転送が最終的に完了した時点の間に、RETRY スレーブは、実行されるすべての転送の試みをチェックし、マスター番号が同じか確認できます。マスター番号が異なることを検知した場合は、以下のような代替行動をとることができます。

- ERROR 応答
- アービタへの信号
- システムレベルの割込み
- システム全体のリセット

3.12.4 分割転送とのバスハンドオーバー

プロトコルは、別のマスターにバスを渡すことができる SPLIT または RETRY 応答の直後にマスターが IDLE 転送を実行するように要求します。図 3-20 では、分割転送の間に発生するイベントを示します。

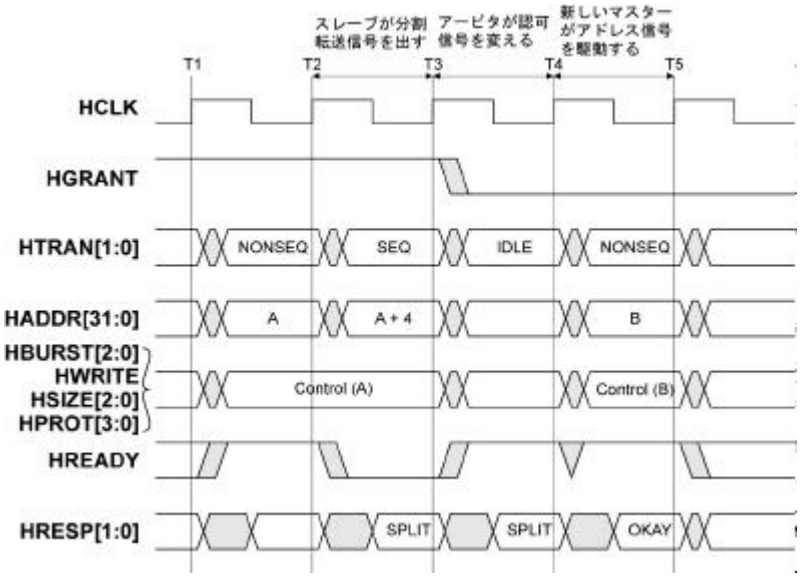


図 3-20 分割転送後のハンドオーバー

以下の点に注意して下さい。

- T1 時点後、転送のアドレスがバス上に現れます。T2 と T3 時点でのクロックエッジの後、スレーブが 2 サイクル SPLIT 応答を返します。
- 最初の応答サイクルの終了時点 T3 で、マスターは、転送が分割されることを検知し、そのため次の転送が IDLE 転送を示すように制御信号を変えることができます。
- T3 時点ではさらに、アービタが応答信号を取得し、転送が分割されたことを確認します。アービタはその後、アービトレーションの優先順位を調整でき、認可信号は次のサイクル時に変わります。その結果、新しいマスターは T4 時点後にアドレスバスを使用できます。
- IDLE 転送が常に 1 サイクルで完了するので、新しいマスターは即時アクセスが保証されます。

3.13 リセット

リセット信号 **HRESETn** は、AMBA AHB 仕様での唯一のアクティブロー信号で、すべてのバスに対する一次リセットです。リセットは非同期にアクティブになることがありますが、**HCLK** の立上がりエッジの後では同期してインアクティブになります。

リセット中、すべてのマスターは、アドレスと制御信号が有効レベルを保ち、**HTRANS[1:0]** が IDLE を示すように保証する必要があります。

3.14 AHB データバス幅について

動作周波数を増加させないで、バス周波数帯域を改良する 1 つの方法は、オンチップバスのデータ経路をより広くすることです。メタル・レイヤーの増強や大型オンチップメモリブロック (埋込み型 DRAM など) の採用は、より広いオンチップバスの使用に役立ちます。

固定されたバス幅を指定することは、多くの場合、バス幅が用途に最適にならないことを意味します。したがって、バス幅の自由度を与え、しかもまだモジュールの設計間で高度な移植性を保証出来るアプローチを採用しました。

プロトコルは、8、16、32、64、128、256、512 または 1024 ビットの AHB データバス幅を見込んでいます。しかし、最小バス幅に 32 ビットを使用することを推奨します。256 ビットという最大幅はほとんどすべての用途に十分であると思われます。

読出し転送と書込み転送の両者に対して、受信モジュールはバス上の正確なバイトレーンからデータを選択しなければなりません。全バイトレーンを通るデータを複製する必要はありません。

3.15 広幅バス上への細幅スレーブの実装

図 3-21 では、本来 32 ビットデータバスと共に動作するように設計されたスレーブモジュールをより広い 64 ビットバス上で動作するように変更するにはどうすればよいかを示します。これは内部的な設計変更よりむしろ、外部のロジックの追加を必要とします。したがって、この手法はハードマクロセルに有効になります。

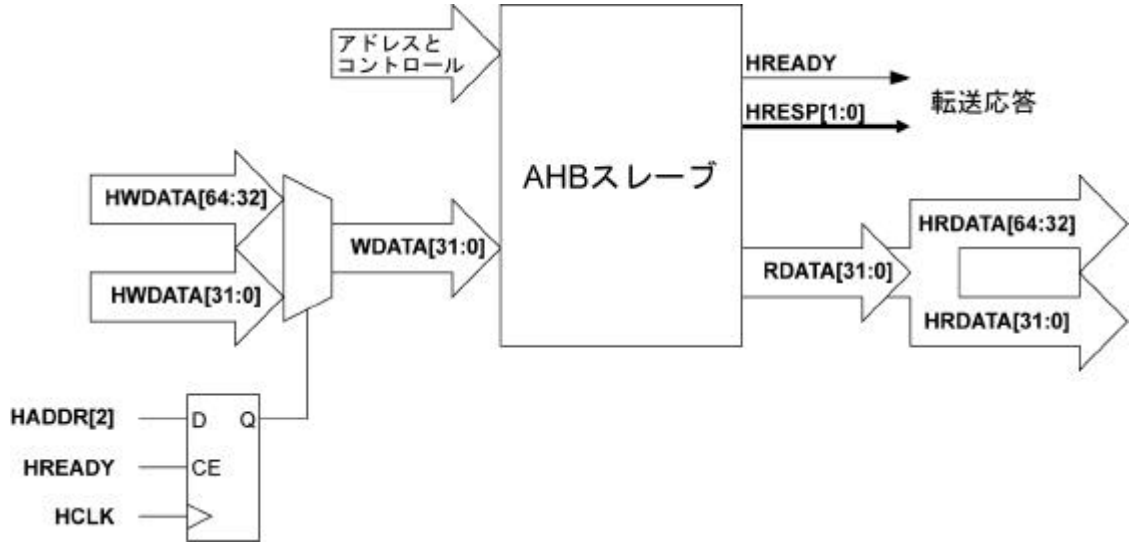


図 3-21 広幅バス上の狭幅スレーブ

出力の場合、狭いバスをより広いバスに変換するときには、以下のうちのどちらかを行なって下さい。

- (上図に示すように) 広いバスを半分にした両側にデータを複製します。
- 追加ロジックを使用して、適切なバスの半分側だけが変化するようにします。これにより電力消費が減少します。

スレーブは、固有のインターフェースと同じ幅の転送だけを受け入れることができます。もしマスターが、スレーブのサポート可能な転送より広い転送を試みた場合、スレーブは ERROR 転送応答を使用できます。

3.16 細幅バス上への広幅スレーブの実装

図 3-22 の例では、狭いバスに実装されている広いスレーブを示します。前と同様、外部のロジックだけが必要です。したがって、事前に設計された、あるいはインポートされたブロックは、異なった幅のデータバスを使用して機能するように容易に修正できます。

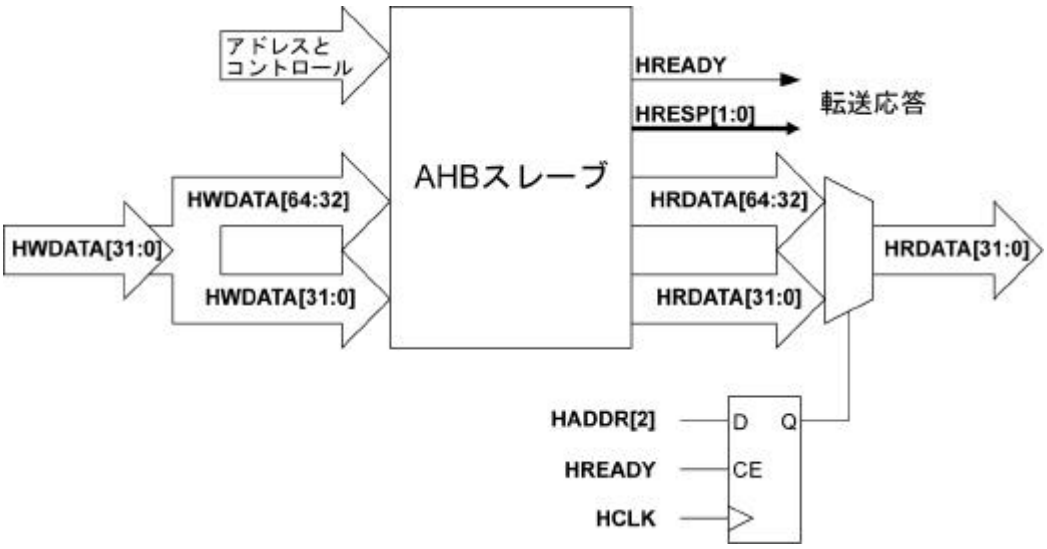


図 3-22 細幅バス上の広幅スレーブ

3.16.1 マスター

バスマスターは、本来意図していたより広いバス上で機能するよう簡単に修正できます。その方法は、より広いバス上で機能するようにスレーブを修正する方法と同じで、以下のとおりです。

- 入力バスの多重化
- 出力バスの複製

しかし、バスマスターが試みる転送の幅を制限するためのメカニズムがマスター内にない場合、そのバスマスターは本来意図していたより狭いバス上で機能するように作成することはできません。マスターは、接続されているデータバスより広い幅 (HSIZE で示される) のところでは転送を試みてはいけません。

3.17 AHB AMBA コンポーネントについて

本セクションでは、AHB ベース AMBA システム内の各回路部について説明し、AMBA 設計の解析に必要な一般的なタイミングパラメータについて説明します。

タイミングパラメータに対して以下の表記法を使用します。

- T_{is} - 入力設定時間
- T_{ih} - 入力保持時間
- T_{ov} - 出力有効時間
- T_{oh} - 出力保持時間

3.18 AHB バススレーブ

AHB バススレーブはシステム内のバスマスターが起動する転送に応答します。本スレーブはデコーダからの HSELx 選択信号を使用して、いつバス転送に応答すべきかを判断します。アドレスと制御情報など、転送に必要な他の信号はすべて、バスマスターが生成します。

3.18.1 インターフェースダイアグラム

図 3-23 には、AHB バススレーブインターフェースを示します。

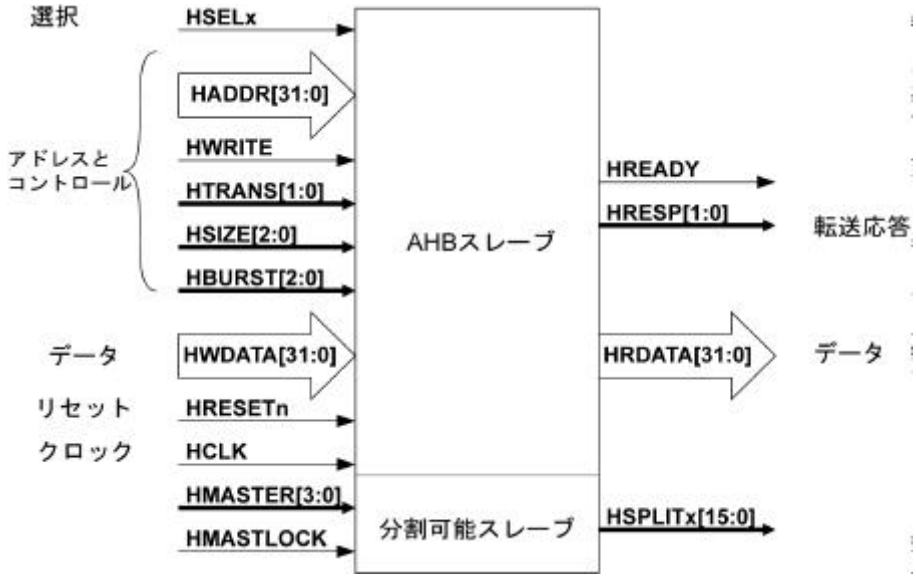


図 3-23 AHB バススレーブインターフェース

3.18.2 タイミングダイアグラム

以下のタイミングダイアグラムは、AMBA システムで動作する AHB バススレーブへのアクセスに関係があるタイミングパラメータを示します。

- 図 3-24 には、AHB スレーブリセットタイミングパラメータを示します。
- 図 3-25 には、メイン AHB スレーブタイミングパラメータを示します。
- 図 3-26 には、分割可能な AHB スレーブの追加タイミングパラメータを示します。

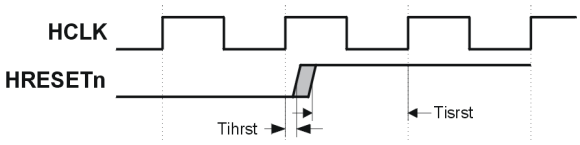


図 3-24 AHB スレーブリセットタイミング

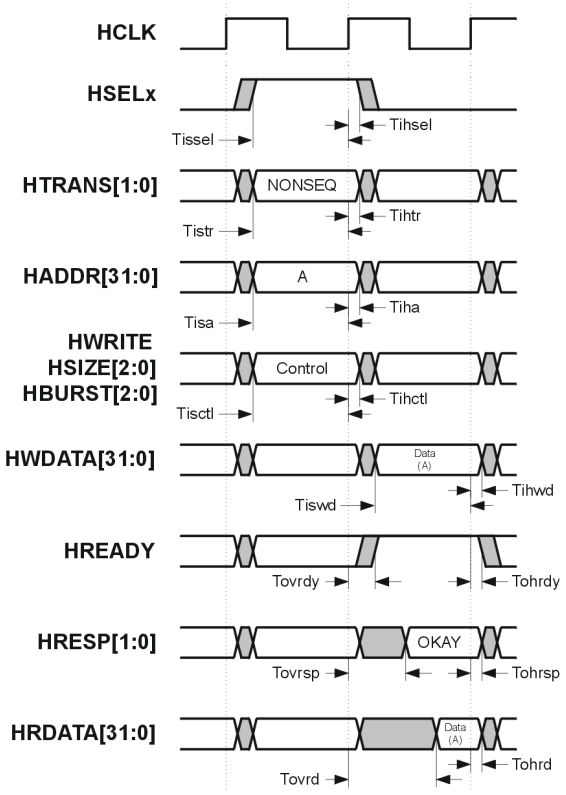


図 3-25 AHB タイミングパラメータ

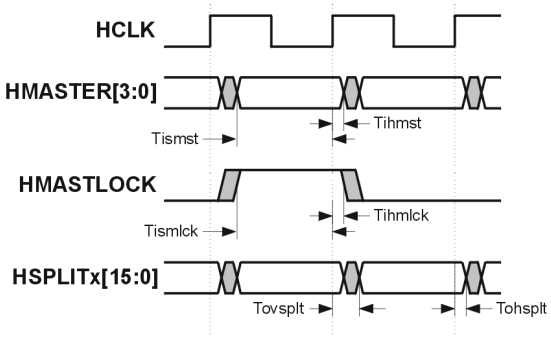


図 3-26 追加分割可能スレーブのパラメータ

3.18.3 タイミングパラメータ

AHB バススレーブに関係があるタイミングパラメータは、入力信号については表 3-81 に、出力信号については表 3-9 に示します。

表 3-8 AHB スレーブ入力パラメータ

パラメータ	説明
T_{clk}	HCLK 最小クロック時間
T_{isrst}	HCLK 前の HRESETn インアクティブ設定時間
T_{ihrst}	HCLK 後の HRESETn インアクティブ保持時間
T_{issel}	HCLK 前の HSELx 設定時間
T_{ihsel}	HCLK 後の HSELx 保持時間
T_{istr}	HCLK 前の転送タイプ設定時間
T_{ihtr}	HCLK 後の転送タイプ保持時間
T_{isa}	HCLK 前の HADDR[31:0] 設定時間
T_{iha}	HCLK 後の HADDR[31:0] 保持時間
T_{isctl}	HCLK 前の HWRITE、HSIZE[2:0]、および HBURST[2:0] 制御信号設定時間

表 3-8 AHB スレーブ入力パラメータ (続き)

パラメータ	説明
T_{ihctl}	**HCLK 後の HWRITE、HSIZE[2:0]、および HBURST[2:0] 制御信号保持時間**
T_{iswd}	**HCLK 前の書込みデータ設定時間**
T_{ihwd}	**HCLK 後の書込みデータ保持時間**
T_{isrdy}	**HCLK 前の設定待ち時間**
T_{ihrdy}	**HCLK 後の保持待ち時間**
T_{ismst}	**HCLK 前のマスター番号設定時間** (SPLIT **可能スレーブのみ**)
T_{ihmst}	**HCLK 後のマスター番号保持時間** (SPLIT **可能スレーブのみ**)
T_{ismlck}	**HCLK 前のマスター固定設定時間** (SPLIT **可能スレーブのみ**)
T_{ihmlck}	**HCLK 後のマスター固定保持時間** (SPLIT **可能スレーブのみ**)

表 3-9 AHB スレーブ出力パラメータ

パラメータ	説明
T_{ovrsp}	**HCLK 後の応答有効時間**
T_{ohrsp}	**HCLK 後の応答保持時間**
T_{ovrdy}	**HCLK 後の有効待ち時間**
T_{ohrdy}	**HCLK 後の保持待ち時間**
T_{ovsplt}	**HCLK 後の分割有効時間** (SPLIT **可能スレーブのみ**)
T_{ohsplt}	**HCLK 後の分割保持時間** (SPLIT **可能スレーブのみ**)

3.19 AHB バスマスター

AHB バスマスターは、AMBA システム内で最も複雑なバスインターフェースを持っています。通常、AMBA システム設計者は既に設計済のバスマスターを使用します。そのためバスマスターインターフェースの詳細に気遣う必要はありません。

3.19.1 インターフェースダイアグラム

AHB バスマスターのインターフェースダイアグラムでは、主な信号グループを示します。

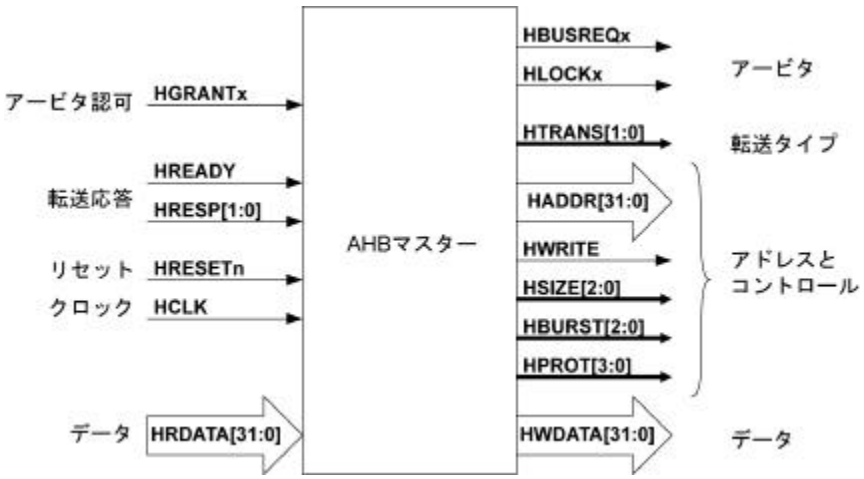


図 3-27 AHB バスマスターインターフェースダイアグラム

3.19.2 バスマスタータイミングダイアグラム

以下のタイミングダイアグラムでは、AMBA システム内で動作する AHB バスマスターと関係があるタイミングパラメータを示します。

- 図 3-28 には、AHB マスターリセットタイミングパラメータを示します。
- 図 3-29 には、AHB マスター転送タイミングパラメータを示します。
- 図 3-30 には、AHB マスターアービトレーションタイミングパラメータを示します。

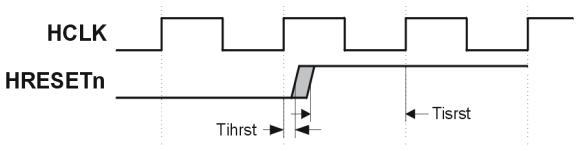


図 3-28 AHB マスターリセットタイミングパラメータ

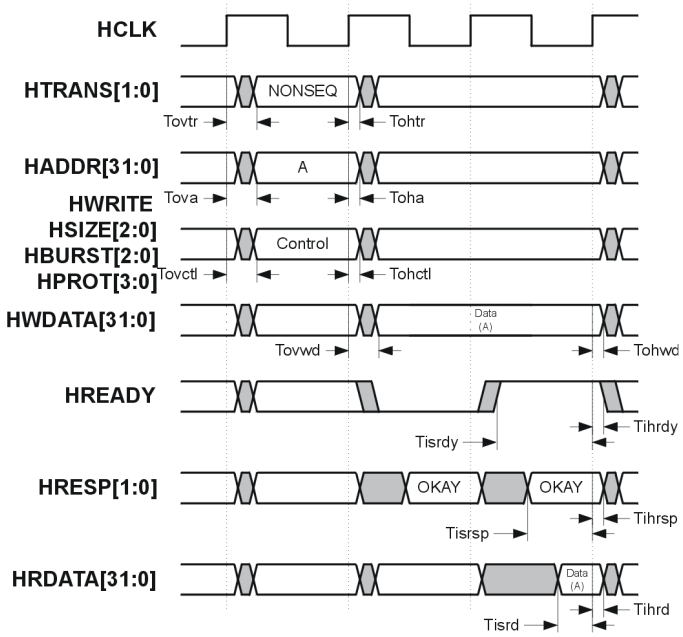


図 3-29 AHB マスター転送タイミングパラメータ

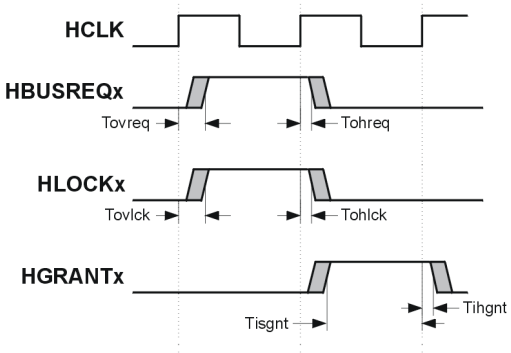


図 3-30 AHB マスターアービトレーションタイミングパラメータ

3.19.3 タイミングパラメータ

AMBA システム内で動作する AHB バスマスターと関係があるタイミングパラメータも、以下の 2 つの表にテキスト形式で示します。表 3-10 には入力信号について、表 3-11 には出力信号について示します。

表 3-10 バスマスター入力タイミングパラメータ

パラメータ	説明
T_{clk}	HCLK 最小クロック時間
T_{isrst}	HCLK 前のリセットインアクティブ設定時間
T_{ihrst}	HCLK 後のリセットインアクティブ保持時間
T_{isgnt}	HCLK 前の HGRANTx 設定時間
T_{ihgnt}	HCLK 後の HGRANTx 保持時間
T_{isrdy}	HCLK 前の設定待ち時間
T_{ihrdy}	HCLK 後の保持待ち時間
T_{isrsp}	HCLK 前の応答設定時間
T_{ihrsp}	HCLK 後の応答保持時間
T_{isrd}	HCLK 前の読出しデータ設定時間
T_{ihrd}	HCLK 後の読出しデータ保持時間

表 3-11 バスマスター出力タイミングパラメータ

パラメータ	説明
T_{ovtr}	HCLK 後の転送タイプ有効時間
T_{ohtr}	HCLK 後の転送タイプ保持時間
T_{ova}	HCLK 後のアドレス有効時間
T_{oha}	HCLK 後のアドレス保持時間
T_{ovctl}	HCLK 後の制御信号有効時間

表 3-11 バスマスター出力タイミングパラメータ（続き）

パラメータ	説明
T_{ohctl}	**HCLK 後の制御信号保持時間**
T_{ovwd}	**HCLK 後の書込みデータ有効時間**
T_{ohwd}	**HCLK 後の書込みデータ保持時間**
T_{ovreq}	**HCLK 後の要求有効時間**
T_{ohreq}	**HCLK 後の要求保持時間**
T_{ovlck}	**HCLK 後のロック有効時間**
T_{ohlck}	**HCLK 後のロック保持時間**

3.20 AHB アービタ

AMBA システム内のアービタの役割は、どのマスターがバスにアクセスできるかを制御することです。すべてのバスマスターはアービタとの REQUEST/GRANT インターフェースを持ち、アービタは優先順位付け機構を使用して、バスを要求しているバスマスターのうち、現在優先順位が最も高いのはどのマスターかを判断します。

各マスターは、バスへの排他的なアクセスを要求していることを示す HLOCKx 信号も生成します。

優先順位方式の詳細は規定されていません。それは用途ごとに定義されます。アービタが他の信号を AMBA か AMBA 以外のどちらかに使用し、使用中の優先順位方式に影響を及ぼしてもかまいません。

3.20.1 インターフェースダイアグラム

図 3-31 では、AHB アービタの信号インターフェースを示します。

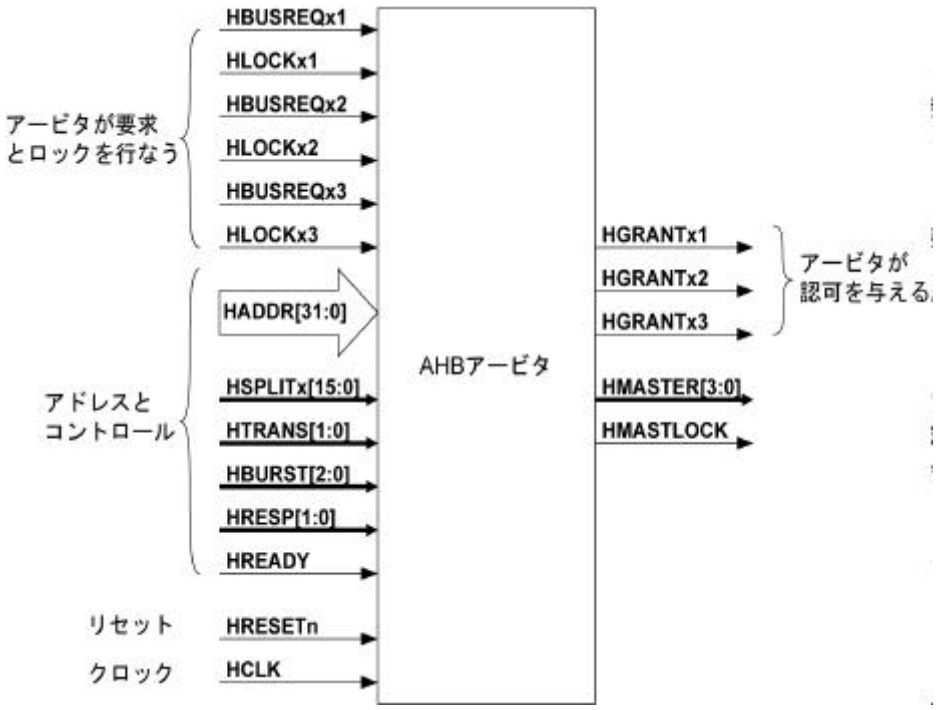


図 3-31 AHB アービタインターフェースダイアグラム

3.20.2 タイミングダイアグラム

以下のタイミングダイアグラムでは、AMBA システム内で動作する AHB バスアービタと関係があるタイミングパラメータを示します。

- 図 3-32 には、AHB アービタリセットタイミングパラメータを示します。
- 図 3-33 には、AHB アービタ転送タイミングパラメータを示します。
- 図 3-34 には、AHB アービタ分割タイミングパラメータを示します。

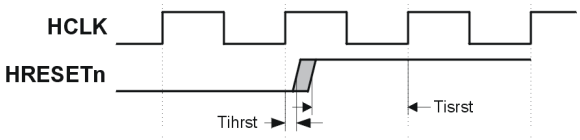


図 3-32 AHB アービタリセットタイミングパラメータ

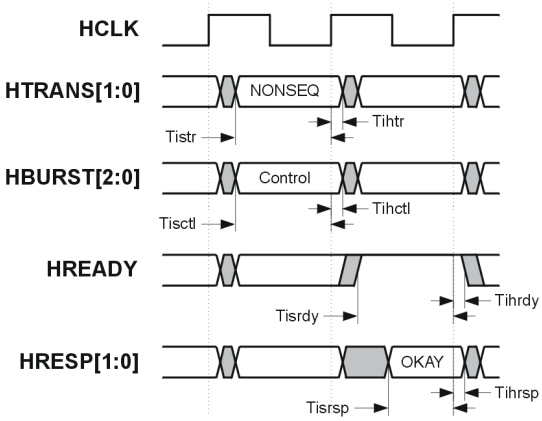


図 3-33 AHB アービタ転送タイミングパラメータ

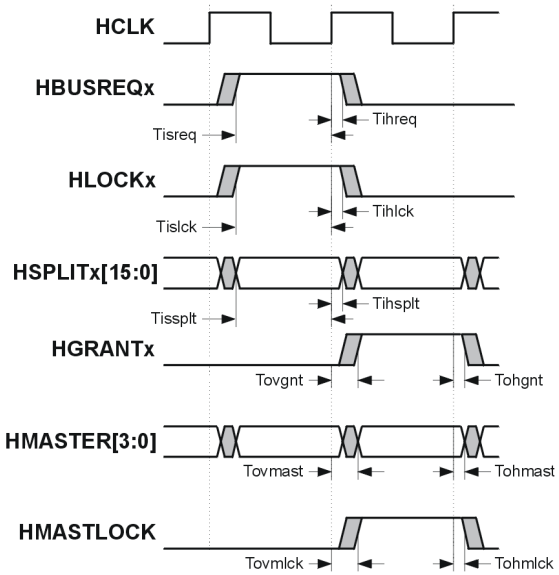


図 3-34 AHB アービタ分割タイミングパラメータ

3.20.3 タイミングパラメータ

AHB アービタと関係があるタイミングパラメータを以下の表に示します。

- 表 3-12 には入力信号について示します。
- 表 3-13 には出力信号について示します。

表 3-12 AHB アービタ入力パラメータ

パラメータ	説明
T_{clk}	HCLK 最小クロック時間
T_{isrst}	HCLK 前のリセットインアクティブ設定時間
T_{ihrst}	HCLK 後のリセットインアクティブ保持時間
T_{isrdy}	HCLK 前の設定待ち時間
T_{ihrdy}	HCLK 後の保持待ち時間
T_{isrsp}	HCLK 前の応答設定時間

表 3-12 AHB アービタ入力パラメータ（続き）

パラメータ	説明
T_{ihrsp}	HCLK 後の応答保持時間
T_{isreq}	HCLK 前の要求設定時間
T_{ihreq}	HCLK 後の要求保持時間
T_{islck}	HCLK 前のロック設定時間
T_{ihlck}	HCLK 後のロック保持時間
T_{issplt}	HCLK 前の分割設定時間
Tihsplt	HCLK 後の分割保持時間
T_{istr}	HCLK 前の転送タイプ設定時間
T_{ihtr}	HCLK 後の転送タイプ保持時間
T_{isctl}	HCLK 前の制御信号設定時間
T_{ihctl}	HCLK 後の制御信号保持時間

表 3-13 AHB アービタ出力パラメータ

パラメータ	説明
T_{ovgnt}	HCLK 後の認可有効時間
T_{ohgnt}	HCLK 後の認可保持時間
T_{ovmst}	HCLK 後のマスター番号有効時間
T_{ohmst}	HCLK 後のマスター番号保持時間
T_{ovmlck}	HCLK 後のマスター固定有効時間
T_{ohmlck}	HCLK 後のマスター固定保持時間

3.21 AHB デコーダ

AMBA システム内のデコーダは、集中型アドレスデコーディング機能を実行するために使用します。この機能は周辺ユニットをシステムのメモリマップから独立したものにすることによって周辺ユニットの移植性を向上します。

3.21.1 インターフェースダイアグラム

図 3-35 では、AHB デコーダを示します。

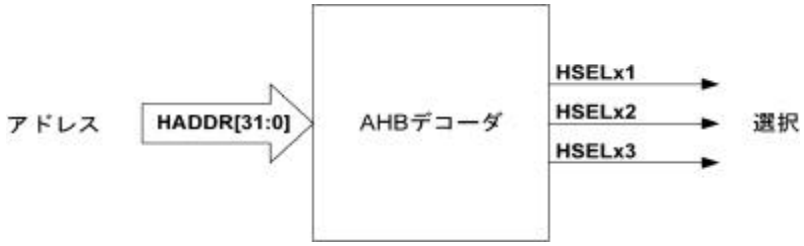


図 3-35 AHB デコーダインターフェースダイアグラム

3.21.2 タイミングダイアグラム

AHB デコーダのタイミングパラメータを図 3-36 に示します。

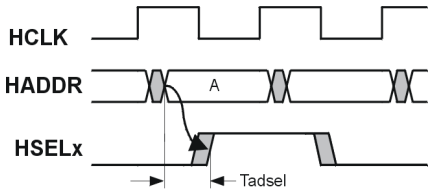


図 3-36 AHB デコーダタイミングパラメータ

3.21.3 タイミングパラメータ

AHB デコーダと関係があるタイミングパラメータを表 3-14 に示します。

表 3-14 AHB デコーダ出力パラメータ

パラメータ	説明
T_{adsel}	アドレスから選択有効までの遅延時間

Chapter 4
AMBA ASB

本章では、*アドバンスド・マイクロコントローラ・バスアーキテクチャ* (AMBA) のアドバンスド・システムバスの仕様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。

4.1 AMBA ASB について

アドバンスド・システムバス (ASB) 仕様書では、高性能 16 ビットおよび 32 ビット組み込み型マイクロコントローラの設計で使用することのできる高性能バスについて規定しています。

AMBA ASB はプロセッサ、オンチップメモリ、および低電力周辺ユニットのマクロセル機能を持ったオフチップ外部メモリインターフェースの接続を効率化します。また、本バスはモジュラーマクロセル試験や診断アクセスのテストインフラストラクチャも提供します。

4.1.1 代表的な AMBA ASB ベース・マイクロコントローラ

AMBA ベース・マイクロコントローラは通常、外部メモリ周波数帯域を維持できる高性能システム *中枢* バスから構成されています。このバスには CPU とその他の *ダイレクト・メモリアクセス* (DMA) デバイスが接続されています。さらに、より低い周波数帯域の周辺ユニットが配置されているより狭幅 APB バスへのブリッジも接続されています。図 4-1 に、代表的な AMBA システムの ASB と APB を示します。

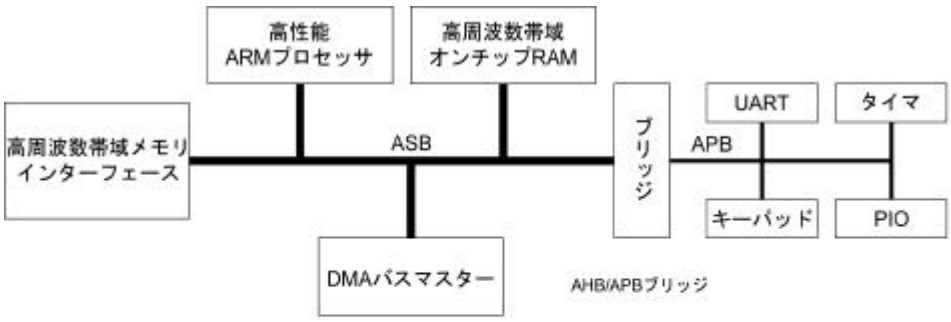


図 4-1 代表的な AMBA システム

外部メモリインターフェースは特定用途向けで、狭いデータ経路しか持っていないこともあるが、テストアクセス・モードをサポートしています。これにより内部 ASB および APB モジュールは、システム独立型テストセットを使用して単独でテストすることができます。

4.1.2 AMBA ASB および APB

APB は、単一の ASB スレーブデバイスとしてまとめられたローカル二次バスとして機能します。APB は、ASB 信号に直接基づいた低消費電力な拡張機能をシステムバスに提供します。

APB ブリッジはローカル周辺バスの代わりに、バスハンドシェイクと制御信号再度タイミング調節を処理するスレーブモジュールとして機能します。システムバスの起点から APB インターフェースを規定することにより、システム診断と試験方法の利点を利用できます。

4.2 AMBA ASB の説明

ASB はパイプライン方式バスで、複数のバスマスターをサポートしています。

そのバスの基本動作フローは以下のとおりです。

1. アービタは、どのマスターにバスへのアクセス権を認可するかを決定します。
2. 認可されると、マスターはバス上で転送を開始します。
3. デコーダは、高位アドレスラインを使用してバススレーブを選択します。
4. そのスレーブは転送応答をバスマスターに返し、マスターとスレーブ間でデータが転送されます。

ASB 上で行なわれる転送タイプは以下の 3 種類です。

NONSEQUENTIAL

単一転送、またはバーストの最初の転送に使用されます。

SEQUENTIAL

バースト内の転送に使用されます。SEQUENTIAL 転送のアドレスは常に前の転送と関係があります。

ADDRESS-ONLY

データの移動が必要でないときに使用されます。ADDRESS-ONLY 転送の 3 つの主用途は、データ転送に関わらない IDLE サイクル、バスマスター HANDOVER サイクル、および推測によるアドレスデコーディングです。

図 4-2 には、バーストトランザクションを行なうための NONSEQUENTIAL および SEQUENTIAL 転送の使用法を示します。

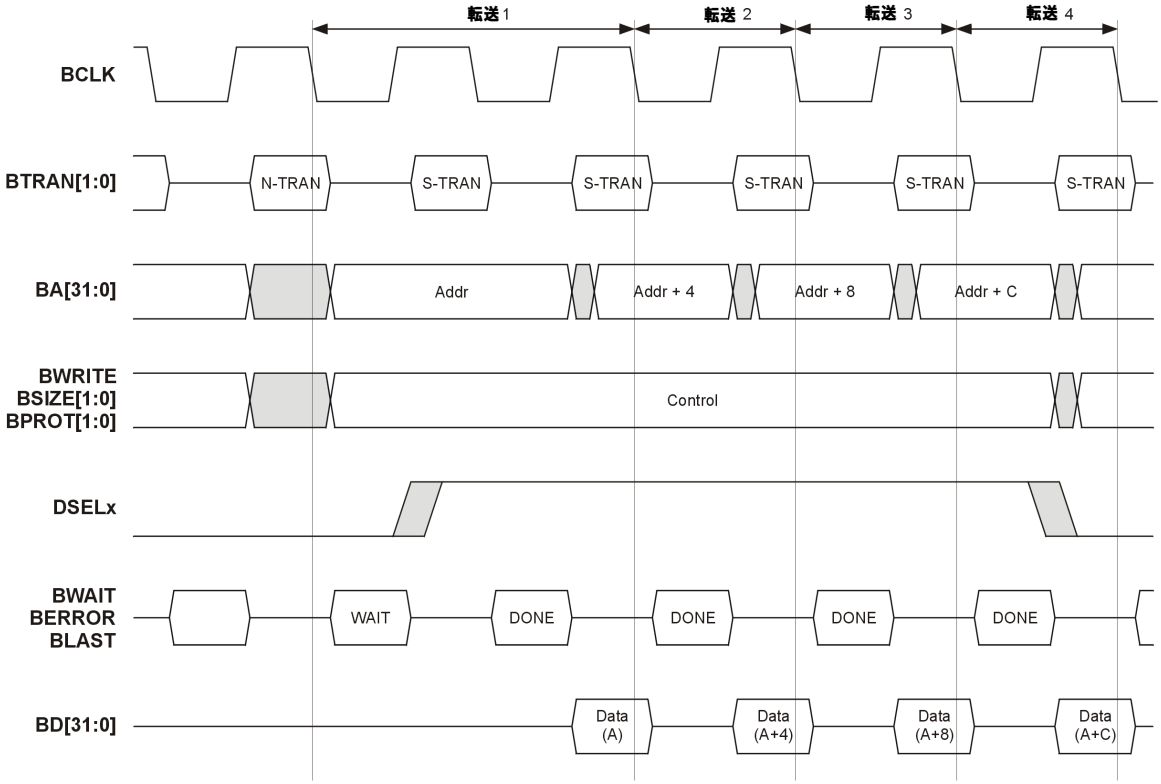


図 4-2 ASB 転送

このバーストは、アドレス A への NONSEQUENTIAL 転送から始まります。後続の SEQUENTIAL 転送は連続したアドレス A+4、A+8、および A+12 に向けたものです。

4.3 ASB 転送

バスの使用が認可されると、マスターは以下の転送を実行できます。

- NONSEQUENTIAL データ転送
- SEQUENTIAL データ転送
- ADDRESS-ONLY 転送

1 つの転送は、**BWAIT** が LOW レベルを示し、その前の転送が完了した後の **BCLK** の立下がりエッジで開始し、再び **BWAIT** が LOW レベルを示し、完全な転送応答を受信した後の **BCLK** の立下がりエッジまで実行されると定義されます。

バスマスターが実行する転送のタイプは、その転送の開始時の **BTRAN** 信号上の値によって判断できます。転送中、**BTRAN** 信号は変化して次の転送のタイプを示します。

4.3.1 ノンシーケンシャル転送

NONSEQUENTIAL 転送は、単一転送に対して、あるいは転送バーストの開始時に行なわれます。図 4-3 には、ウェイトステートを含んでいる代表的な NONSEQUENTIAL 読出し転送を示します。

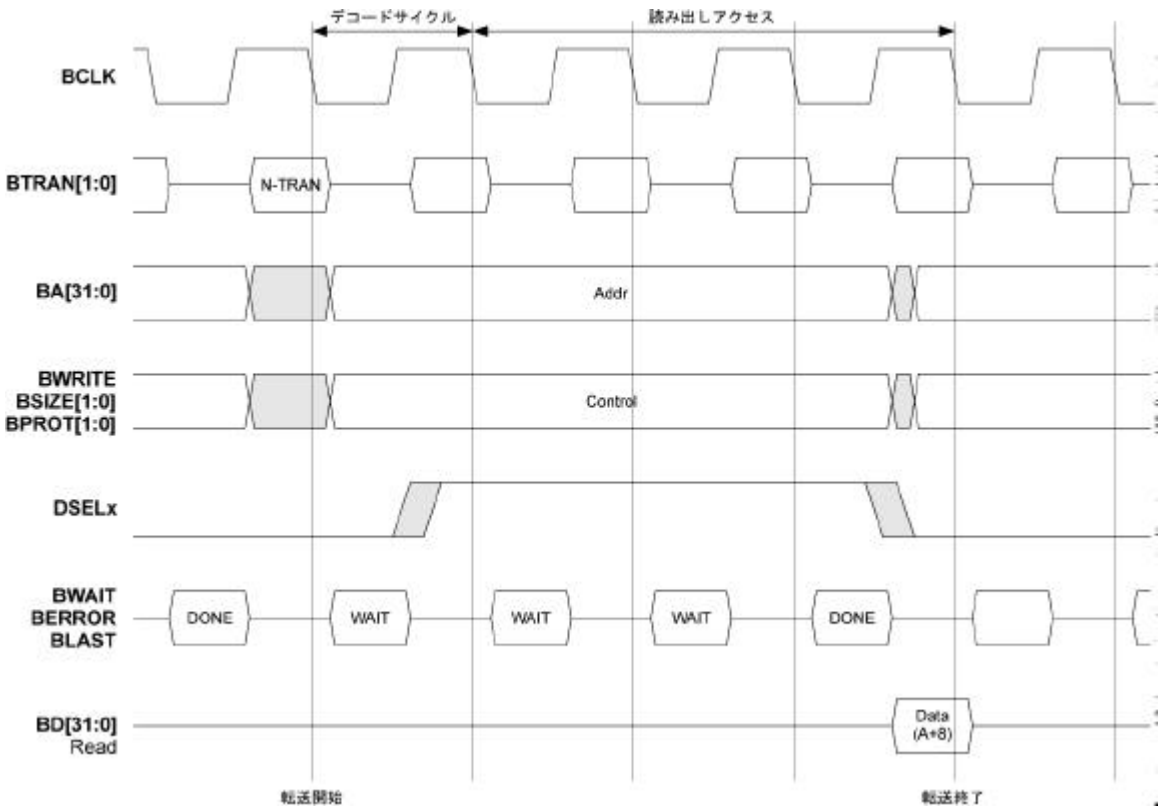


図 4-3 ノンシーケンシャル転送

以下の点に注意して下さい。

- アドレスおよび制御信号は、転送が始まる前の **BCLK** HIGH フェーズで変化し始めます。
- NONSEQUENTIAL 転送の場合、**BCLK** HIGH フェーズの最後の時点まで、あるいは転送の初めのクロック LOW フェーズの開始まで、有効なアドレスが入手できない場合があります。

- 正しいスレーブを選択するために安定したアドレスを必要とするデコーダは、NONSEQUENTIAL 転送の最初のサイクルにウェイトステートを自動的に挿入します。これは DECODE サイクルと呼ばれ、これによってデコーダが DECODE サイクルの HIGH フェーズ中に高順位アドレスラインを検査し、該当する **DSELx** をアクティブにするのに十分な時間が確保されます。
- 転送の残りのサイクルの場合は、スレーブが転送応答を返し、そしてマスターとスレーブ間でデータ交換が行なわれます。

Note

あるシステム設計、通常は低周波数システムクロックを搭載したシステム設計では、アドレスは転送開始前の **BCLK** HIGH フェーズの初期時点で有効になります。このためデコーダは **BCLK** の立下がりエッジの前に有効 **DSELx** 信号を生成できます。このようなシステムは NONSEQUENTIAL 転送の開始時に DECODE サイクルの追加を必要としません。このようなシステムの動作の詳細については、P.4-14 *「アドレスデコード」*を参照して下さい。

- データバス **BD[31:0]** は、転送終了時の **BCLK** の立下がりエッジ付近で有効でなければなりません。書込みサイクル中、バスマスターはデータバスを駆動します。これは、スレーブがクロックの立下がりエッジ付近で有効データを受け取るために、クロック HIGH フェーズの開始から行なわれます。読出しサイクル中は、HIGH フェーズの終了近くで有効であるように該当するスレーブがデータバスを駆動しなければなりません。
- 多くの様々なバススレーブがデータを ASB に送るので、データをバスに送るスレーブがオーバーラップしないことを保証する必要があります。NONSEQUENTIAL 転送開始時のクロックが LOW の間は、スレーブとマスターがデータを送らないので、1 つのノンオーバーラップへの全フェーズが提供されます。
- バス信号の多くは共用され、アクティブなドライバが存在していないターンアラウンド時間を持ちます。フローディングのレベルがバス上に現れるのを防ぐために、バスホールドのセルを提供しなければなりません。

4.3.2 シーケンシャル転送

アドレスが前の転送のアドレスと関係がある場合、SEQUENTIAL 転送が行なわれます。**BWRITE**、**BPROT**、および **BSIZE** が示す制御情報は、前の転送と同じです。

SEQUENTIAL 転送が NONSEQUENTIAL 転送または別の SEQUENTIAL 転送の後に続く場合、そのアドレスは前のサイズとアドレスを使用して計算できます。例えば、ワードアクセスのバーストはアドレス A、A+4、A+8 向けであり、ハーフワード・アクセスのバーストはアドレス A、A+2、A+4 向けです。

SEQUENTIAL 転送が ADDRESS-ONLY サイクルの後に続く場合、そのアドレスは ADDRESS-ONLY サイクルのアドレスと同じです。ADDRESS-ONLY 転送、続いて SEQUENTIAL 転送というこの組合わせにより、SEQUENTIAL 転送を使用したアクセスが可能になるだけでなく、転送バーストが SEQUENTIAL 転送で開始できます。ADDRESS-ONLY 転送、続いて SEQUENTIAL 転送という使用例を図 4-6 に示します。

図 4-4 には、1 つのウェイトステートを挿入した SEQUENTIAL 転送を示します。これは NONSEQUENTIAL 転送とよく似ています。

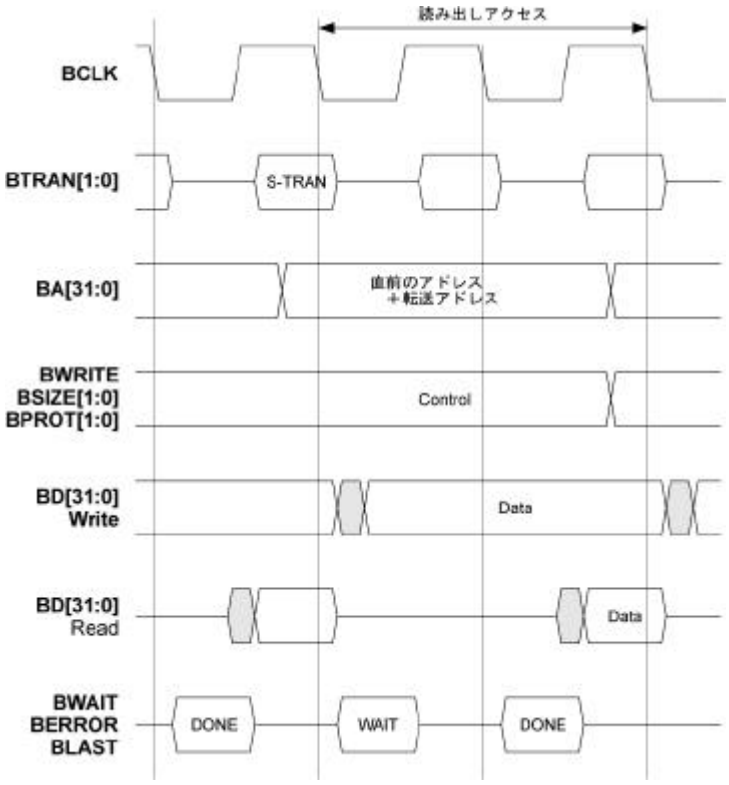


図 4-4 シーケンシャル転送

主な相違点は以下の通りです。

- **BTRAN** 信号が SEQUENTIAL 転送を示します。
- 転送開始時の **BCLK** HIGH フェーズでアドレスが常に有効です。
- アドレスが直前の転送と関係があります。

- 制御情報が直前の転送と同じままです。
- 書込みの場合、転送中ずっとデータバスが駆動されます。

NONSEQUENTIAL 転送の場合と異なり、バス・ターンアラウンドにかかる時間を提供する必要がないので、データバス **BD[31:0]** は転送中ずっと駆動することができます。

4.3.3 アドレスオンリー転送

ADDRESS-ONLY 転送は、データトランザクションが必要ないことを示します。ADDRESS-ONLY 転送中は、アドレスおよび制御信号が無効の場合があります。有効レベルにまで駆動しなければならない信号は以下の信号だけです。

- **BTRAN** - 次の転送のタイプを示します。
- **BLOK** - アービトレーション処理を続行できるようにします。

ADDRESS-ONLY トランザクションはバス上のスレーブにアクセスしないので、1 サイクルしか必要ありません。したがって、**BWAIT** 信号は LOW になります。ADDRESS-ONLY サイクル中にスレーブが選択されないので、この信号はバスデコーダによって駆動されます。データ転送用バスを必要としない場合、バスマスターは多くの ADDRESS-ONLY 転送を連続的に実行します。

ADDRESS-ONLY 転送は、以下の 3 つの方法で使用できます。

- 真の IDLE サイクルとして (バスマスターがバスを必要としないとき)。
- 転送に専念することなく、次の転送のアドレスを推測して送信すること。
- バスマスター・ハンドオーバー中にターンアラウンド・サイクルを与えること。

ADDRESS-ONLY 転送が、真の IDLE サイクルとして使用されている場合、アドレスおよび制御信号はその転送中のどこでも有効にする必要はありません (図 4-5 参照)。

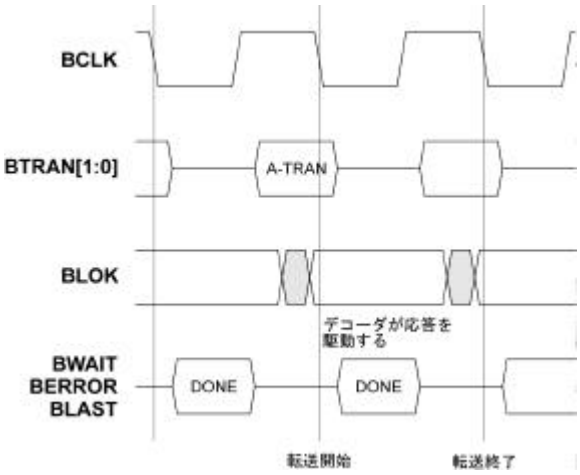


図 4-5 アドレスオンリー転送

BLOK 信号は唯一の例外で、この信号はアービトレーションプロセスを続行するために、全 ADDRESS-ONLY 転送中に有効レベルまで駆動されなければなりません。

ADDRESS-ONLY 転送の 2 番目の使用法は、転送に専念せずに、ある転送のアドレスを推測して送信することです (図 4-6 参照)。

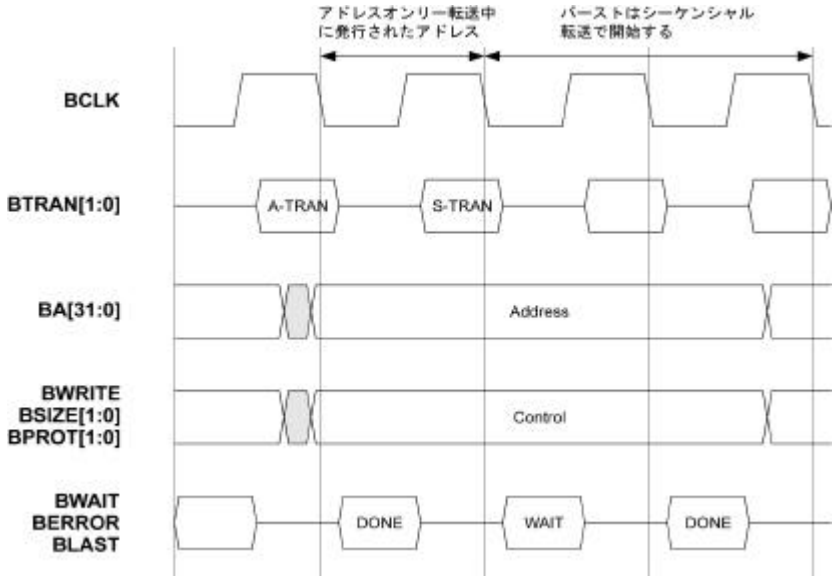


図 4-6 バーストを開始するためのアドレスオンリー転送

アドレスを推測して送信するために ADDRESS-ONLY 転送を使用すると、デコーダは ADDRESS-ONLY サイクル中にアドレスデコーディングで実行できるようになります。バスマスターがバーストの実行を約束した場合は、そのバーストを SEQUENTIAL 転送で開始することができ、そのため、転送の開始前の余分な DECODE サイクルが必要でなくなります。

ADDRESS-ONLY の最後の使用法は、バスマスター・ハンドオーバー中にターンアラウンド時間を提供することです (図 4-7 参照)。

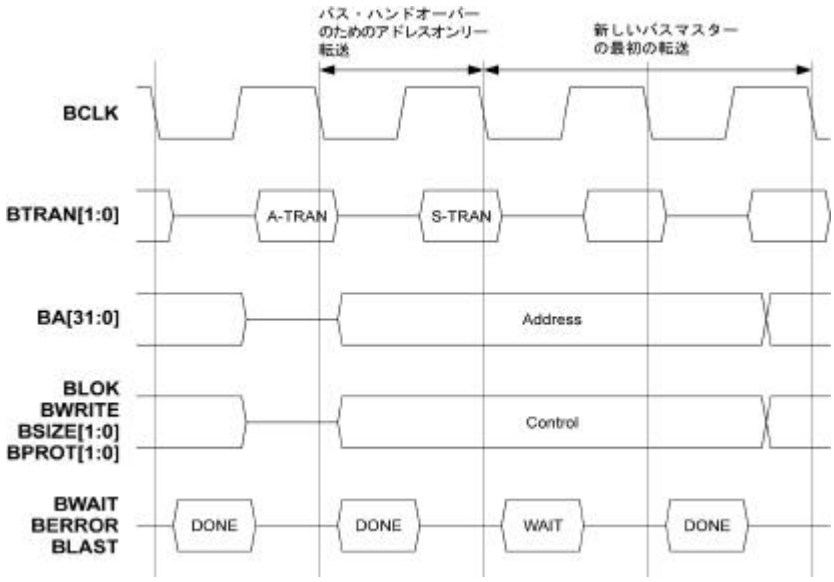


図 4-7 バスマスター・ハンドオーバーのためのアドレスオンリー転送

バスの使用を認可されたバスマスターは、ADDRESS-ONLY 転送で開始します。この場合、その新しいバスはアドレスと制御信号をすぐには駆動しないが、その転送の LOW フェーズにおいてそれらの信号を駆動する前に、ターンアラウンド・フェーズを設けます。

Note

この場合、アドレスおよび制御情報は、**BCLK** の LOW フェーズまで有効にはなりません。

4.4 アドレスデコード

ASB ベース AMBA システムでは、アドレスデコーディングは集中型デコーダによって行なわれます。

デコーダは各転送タイプを使用して、以下の機能のうちのどれを実行すべきかを決めます。

- ADDRESS-ONLY 転送の場合、デコーダが DONE 転送応答で応答し、スレーブは選択されません。ADDRESS-ONLY 転送中に、ADDRESS-ONLY 転送のすぐ後に SEQUENTIAL 転送が続く場合、デコーダは推測によりアドレスデコードを実行します。
- NONSEQUENTIAL 転送の場合 (または前の転送が LAST トランザクション応答で終った場合)、デコーダは転送開始時に、1 回のウェイトステートを挿入してアドレスデコーディングのために十分な時間を確保します (この追加のウェイトステートがすべてのシステムに必要とは限りません)。

  デコーダが挿入したこのウェイトステートは、*DECODE サイクル*と呼ばれ、DECODE サイクル中は、選択信号 **DSELx** はアクティブにはなりません。

  転送の 2 番目のサイクルでは、デコーダは適切なスレーブを選択するか、ERROR 転送応答を返します。

  ERROR 応答は、以下の状況で返されます。

  - 転送のアドレスにスレーブが存在しない。
  - 転送が保護されたメモリ領域に対するものである。
  - 転送の調整がメモリシステムによってサポートされていない。

  非常に一般的な有効な転送の場合では、デコーダは該当するスレーブの **DSELx** 信号をアクティブにし、そして選択されたスレーブが転送の残りサイクルについて転送応答を返すことを可能にします。
- SEQUENTIAL 転送の場合、デコーダは該当する **DSELx** 信号をアクティブにし、選択されたスレーブは転送応答を返します。前の NONSEQUENTIAL または ADDRESS-ONLY 転送で実行されているので、デコーダがアドレスをデコードする必要はありません。

  SEQUENTIAL 状態において、デコーダはアドレスデコードを行なわないので、転送がメモリ境界を超えようとしている場合、スレーブが LAST 転送応答を返す必要があります。また、デコーダは、SEQUENTIAL 転送がメモリ境界を超えるであろうと判断したとき、**LAST** 信号を内部向けに生成しなければなりません。

NONSEQUENTIAL 転送時の DECODE サイクルの挿入は、システムの性能を向上するために使用できます。ある特有の設計において、アドレスデコーディングに必要な時間はスレーブへのアクセスの危険なパスを増加させ、追加のウェイトステートが必要になることが度々あります。デコーダはこのオーバーヘッドを低減するために使用でき

ます。低減するためには、NONSEQUENTIAL 転送時にのみ DECODE サイクルを自動的に挿入しますが、SEQUENTIAL 転送が追加のウェイトステート無しで完了できるようにして下さい。

一部のシステム、通常の低クロック周波数を搭載したシステムでは、アドレスデコーディングのために追加のウェイトステートは必要ありません。このようなシステムでは、デコーダは簡単になり、その結果、SEQUENTIAL および NONSEQUENTIAL 転送は両者とも、DECODE サイクルの追加無しで行なわれます。

4.5 転送応答

バスマスターによって開始される全ての転送のために、応答が生成され、この応答はデコーダか選択されたバススレーブかどちらかによって返されます。この転送応答は、クロック LOW フェーズ時に駆動される **BWAIT**、**BERROR**、および **BLAST** 信号を使って返されます。

図 4-8 には、転送を延長する目的で 3 つのウェイトステートを挿入する際の転送応答の使用法の実例を示します。

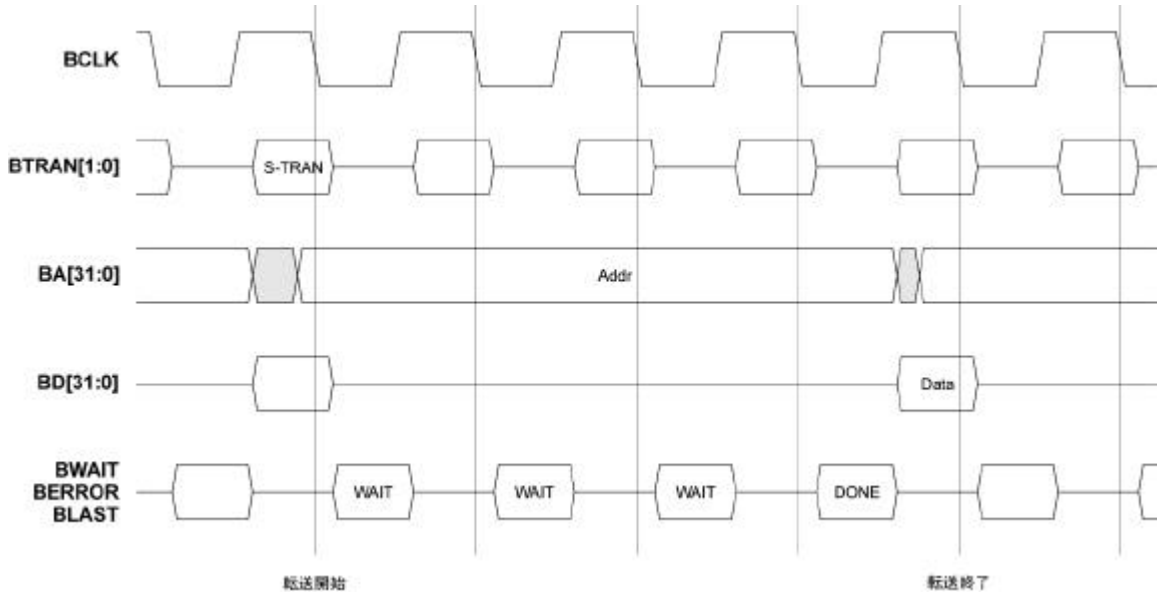


図 4-8 転送応答

以下の転送応答が使用可能です。

WAIT　転送が完了する前に、その転送を延長しなければなりません。

DONE　転送が成功した上で完了しました。

ERROR　転送は完了したが、エラーが発生しました。このエラー状態はバスマスターに信号で通知されます。その結果、バスマスターはその転送が成功しなかったことを知ります。

LAST　転送は成功した上で完了したが、スレーブがこれ以上のバースト転送を受け入れることができないか、あるいはメモリ境界に到達しました。この応答は、バスマスターに対する DONE と同じです。しかし、これは次の転送の開始時に DECODE サイクルを挿入する必要があることをデコーダに知らせます。

RETRACT　転送がまだ完了していません。したがって、バスマスターは転送を再試行すべきです。RETRACT 応答は、転送は完了出来るが、多くのバスサイクルを必要となり得る事を、スレーブを使ってバスマスターに知らせます。

RETRACT 応答を使用すると、完了するのに長時間かかるかもしれない転送によって、バスがロックされるのを防止し、また優先順位がより高いバスマスターが使用できるようにバスを解放できます。

1 サイクルかかる他の応答コードとは異なり、RETRACT 応答は図 4-9 に示すように、２ステージの応答です。

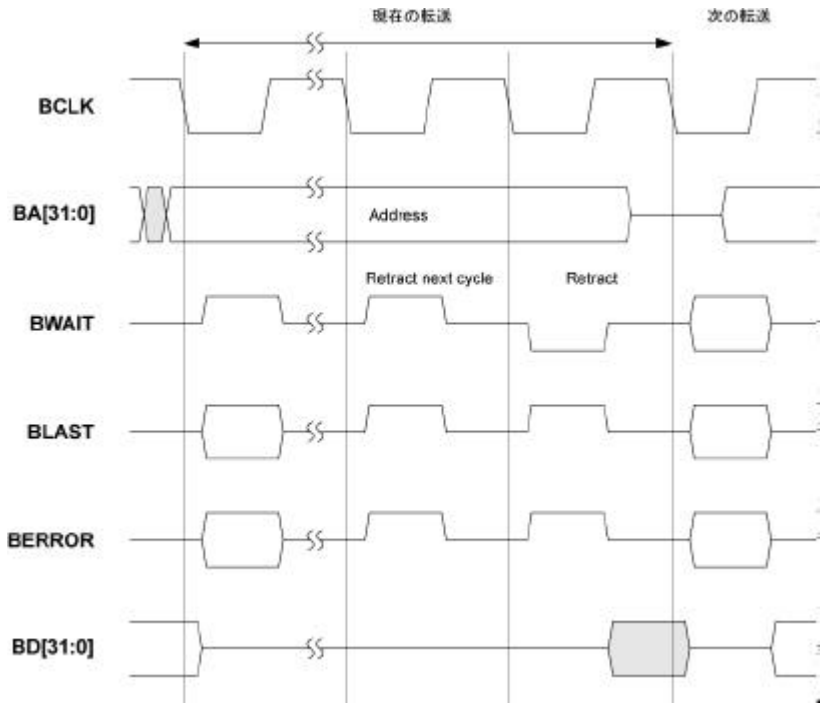


図 4-9 リトラクト動作

以下の点に注意して下さい。

- 第 1 ステージでは、スレーブは RETNEXT 応答 (**BWAIT**、**BLAST**、および **BERROR** がすべて HIGH) を使用して、今にも RETRACT が発生しそうであることをバスマスターに知らせます。
- 第 2 ステージでは、スレーブが RETRACT 応答 (**BWAIT** が LOW で、**BLAST** と **BERROR** の両者が HIGH) を返したとき、転送が完了します。

この 2 ステージ応答は、**BWAIT** 信号が LOW になった時点で転送が完了していると判断してはいけない、という十分な警告をバスマスターに与えます。

バスマスターはすべて、RETRACT メカニズムに対応しなければなりません。しかし、スレーブすべてが RETRACT 応答の実施を必要としているわけではありません。通常、RETRACT 応答は、短い確保された完了時間を持たないので、かなりの時間バスをデッドロック状態にするスレーブによって返されるだけです。

ほとんどの転送の場合、この応答は選択されたバススレーブによって返されます。しかしこのような時は、デコーダが応答します。

- 転送が ADDRESS-ONLY である。
- 転送がバススレーブの無いメモリ領域用である。
- 保護メモリ領域へのアクセス違反が発生する。

4.6 マルチマスター動作

AMBA バス仕様は、高性能 ASB 上の複数のバスマスターに対応しています。単純な 2 線式要求および認可メカニズムが、アービタとバスマスター間に実装されています。アービタは、バス上のアクティブなバスマスターが 1 つだけであることを保証し、さらにバスを要求しているマスターが無いときに、デフォルトマスターにバスの使用が認可されることを保証します。

AMBA バス仕様は、共用ロック信号にも対応しています。これによりバスマスターは、現在の転送を次の転送から分離すべきでないことを示すことができます。さらに、ロックされた転送が完了する前に、他のバスマスターがバスへのアクセス権を取得するのを防止します。

バス上のアクティブな連続したマスター間の *デッドタイム*を低減するためには、効率的なアービトレーションが重要です。バスプロトコルはパイプライン方式アービトレーションをサポートしています。そのため、次の転送についてのアービトレーションは現在の転送中に行なわれます。

アービトレーションプロトコルは規定されていますが、優先順位付けは柔軟で、そのアプリケーションにまかされています。しかし、*テストインターフェース*は通常、あらゆる条件下でのテストアクセスを確保するため、最も高い優先順位がつけられています。さらに、すべてのシステムは、バスを要求しているバスマスターが無いときにバスの使用が認可されるデフォルト・バスマスターを組み込んでいます。

各バスマスターからアービタへの要求信号 **AREQx** は、そのバスマスターがバスを必要としていることを知らせます。アービタからバスマスターへの認可信号 **AGNTx** は、そのバスマスターが現在、バスを要求しているマスターのうちで優先順位が最も高いことを知らせます。

バスマスターは、以下のように動作します。

- **AGNTx** が HIGH の場合、**BCLK** HIGH 中に **BTRAN** 信号を駆動しなければなりません。
- **BCLK** の立上がりエッジで **AGNTx** が HIGH、かつ **BWAIT** が LOW の場合、バスの使用を認可されます。

共用バスロック信号 **BLOK** は、次の転送が現在の転送と不可分で、他のどのバスマスターにもバスへのアクセス権を与えるべきでないことをアービタに知らせます。

バスマスターは、バスの使用を認可された時は必ず **BLOK** 信号を有効にして、そのバスマスターが転送を行なっていなくても、アービトレーションプロセスが続行できるようにします。

4.6.1 アービタ

アービタの機能は以下の通りです。

1. バスマスターは、**BCLK** の HIGH フェーズ中に **AREQx** をアクティブにします。
2. アービタは、**BCLK** の立下がりエッジ時にすべての **AREQx** 信号を取得します。
3. **BCLK** の LOW フェーズ中、アービタは **BLOK** 信号を取得し、適切な **AGNTx** 信号をアクティブにします。
   - **BLOK** が LOW の場合、アービタは優先順位が最も高いバスマスターにバスの使用を認可します。
   - **BLOK** が HIGH の場合、アービタは同じバスマスターを認可された状態で保持します。

アービタは、バスサイクルごとに認可信号を更新できます。しかし、新しいバスマスターは、**BWAIT** が LOW になり現在の転送が完了した時、バス駆動権を与えられ、開始できるようになるだけです。したがって、動作可能な次のバスマスターが転送待機中に変化することがあります。

BLOK 信号は、2 つの異なったバスマスター間のハンドオーバーの 1 サイクル中、アービタによって無視されます。

どのバスマスターもバスを要求していない場合、アービタはデフォルト・バスマスターを認可しなければなりません。

アービトレーションプロトコルは規定されていますが、優先順位付けは柔軟で、そのアプリケーションにまかされています。単純な固定優先順位方式を使用している場合もあります。もう一つの方法として、そのアプリケーションが必要とするなら、もっと複雑な方式も実装可能です。

4.6.2 バスマスター・ハンドオーバー

バス使用権を与えられていないバスマスターが新しい認可バスマスターになると、バスマスター・ハンドオーバーが生じます。

AGNTx が HIGH、かつ **BWAIT** が LOW のとき、バスマスターがバスの使用を認可します。**AGNT** の HIGH 状態は、そのバスマスターが、バスを要求しているマスターのうちで優先順位が最も高いことを示し、**BWAIT** LOW は前の転送が完了したことを示します。

図 4-10 には、バスマスター・ハンドオーバー・プロセスを示します。

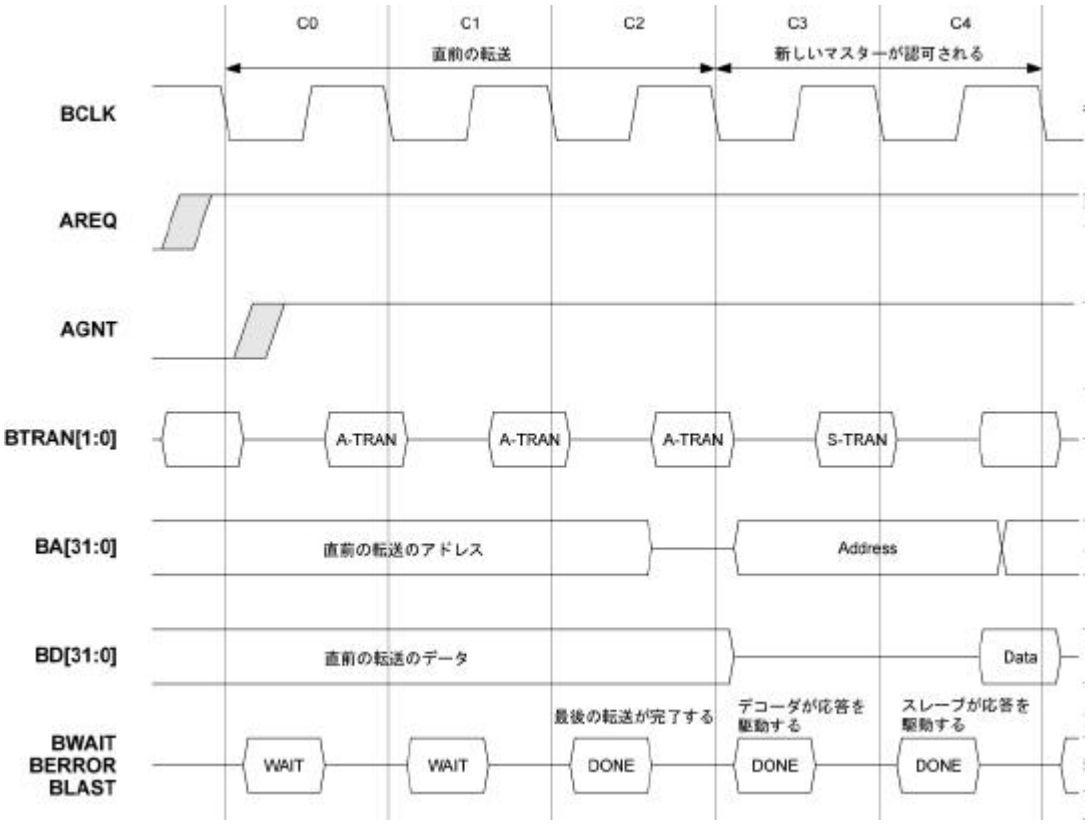


図 4-10 バスマスター・ハンドオーバー

以下の点に注意して下さい。

- **AGNTx** がアクティブになると、**BCLK** HIGH 中にバスマスターが **BTRAN** 信号を駆動します。これは、前の転送が後回しになっている場合、多くのサイクルで継続することがあります。
- バス・ターンアラウンドを見込むために新しいバスマスターが ADDRESS-ONLY サイクルで開始しなければならないので、ハンドオーバーに先立って、**BTRAN** が ADDRESS-ONLY サイクルを示す必要があります。
- 前の転送が完了すると、新しいバスマスターが認可されます。
- 前の転送の最終クロック HIGH フェーズで、アドレスバスは前のバスマスターによる駆動を停止します。

- クロック LOW フェーズ中、新しいバスマスターはアドレスおよび制御信号を駆動し始めます。その後、次のバスサイクルで、最初の転送が始まります。

転送がウェイトステートにある間、バスマスター・ハンドオーバーは遅れることがあります。現在の転送が完了する前に、もう一つのより高い優先順位にあるバスマスターがバスを要求した場合、特定のバスマスターへの **AGNTx** がアクティブを要求し、取り消されることがあり得ます。

4.6.3 デフォルト・バスマスター

システムはすべて、他のどのバスマスターもバスを要求していないときに、認可された唯一のデフォルト・バスマスターを組み込むように設計されています。デフォルト・バスマスターは、バスを確実にアクティブ維持するのに、以下の信号を駆動します。

- ADDRESS-ONLY 転送を示すために **BTRAN** を駆動しなければなりません。
- **BLOK** を LOW にしなければなりません。

4.6.4 ロックされた転送

バスマスターが、RETRACT 応答を返せるスレーブに対して、ロックされた転送の実行を試みないというのは大切な事柄です。この理由には次の 2 つがあります。

- 非常に多くのサイクルの間、バスがロック維持します。
- ロックされた転送シーケンスの最後の転送時に RETRACT が行なわれた場合、その転送が完了する前にアービタはバスの所有権を次のマスターに切り換えられています。従って、最終転送はシーケンスにロックされていないことになります。

4.7 リセット動作

リセット信号 **BnRES** はアクティブロー信号で、バスが安全な状態にあることを保証するのに、非同期にアクティブにすることができます。リセット中は、以下の動作がバス上で行なわれます。

- アービタがデフォルト・バスマスターにバスの使用を認可します。
- デフォルト・バスマスターは、以下の動作を行ないます。
  - **BTRAN** を駆動して、ADDRESS-ONLY 転送を示します。
  - **BLOK** を LOW にしてアービトレーションを可能にします。
- 他のすべてのバスマスターが共用バス信号をトライステートにします。
- デコーダは、以下の動作を行ないます。
  - スレーブ選択信号 **DSELx** をすべてインアクティブにします。
  - 当該転送応答を返します。
- すべてのスレーブが共用バス信号をトライステートにします。

4.7.1 リセットの終了

図 4-11 には、リセットシーケンスの終了を実例で示します。

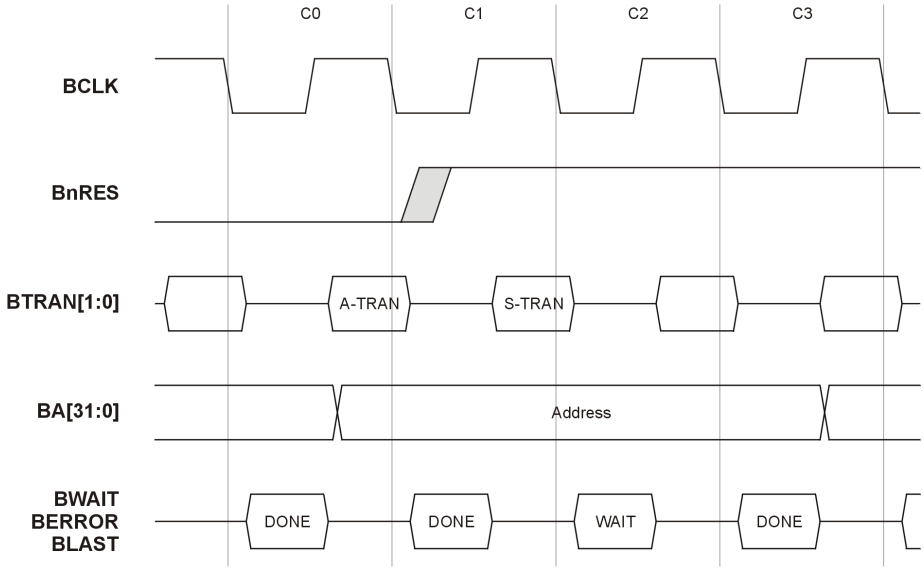


図 4-11 リセットの終了

以下の点に注意して下さい。

- サイクル C1 中は、クロック LOW 時に **BnRES** がインアクティブになります。

- サイクル C1 のクロック HIGH フェーズ中、デフォルト・バスマスターは **BTRAN** 信号を駆動して転送の開始要望をしていることを示します。
- サイクル C2 中に転送が開始します。本例では、転送が後回しにされ、サイクル C3 に進みます。

4.8 ASB 信号の説明

本セクションでは、すべての AMBA ASB 信号について、それらの用途やフェーズに正確なタイミング必要条件を含んでもっと詳細に説明します。

フローティングレベルがバス上に存在するのを防ぐために、バス保持セルを提供しなければなりません。この理由は、バス信号の多くが共用され、アクティブドライバが無い時のターンアラウンド時間を持っているからです。

4.8.1 クロック

BCLK は、すべてのバス転送のタイミングをとるのに使用される一次クロックです。このクロックの両エッジが使用されます。

4.8.2 リセット

ただ 1 つのアクティブロー・リセット信号 **BnRES** がサポートされており、これはバスをリセットするために使用されます。リセット信号は、どちらかのクロックフェーズ中に他の信号と非同期に LOW になりますが、**BCLK** の LOW フェーズ中は常にインアクティブになっています。

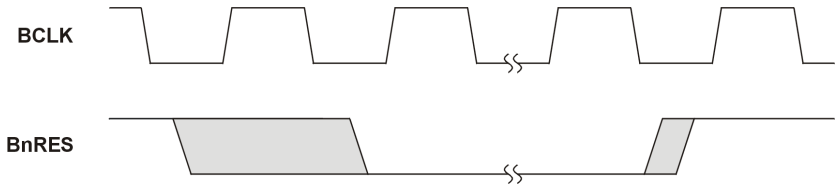


図 4-12 リセット信号

リセット中は、以下の動作がバス上で行なわれます。

- アービタがデフォルト・バスマスターにバスの使用を認可します。
- デフォルト・バスマスターは、以下の動作を行ないます。
  - **BTRAN** を駆動して、ADDRESS-ONLY 転送を示します。
  - **BLOK** を LOW にしてアービトレーションを可能にします。
- 他のすべてのバスマスターが共用バス信号をトライステートにします。
- デコーダは、以下の動作を行ないます。
  - スレーブ選択信号 **DSELx** をすべてインアクティブにします。
  - 適切な転送応答を返します。
- すべてのスレーブが共用バス信号をトライステートにします。

BnRES 信号は、タイムアウト状態中に、バスをリセットするために使用してもかまいません。

ほとんどのバスマスターおよびスレーブにおいて、**BnRES** 信号はそのコンポーネントのバスインターフェースとメインコアの両方をリセットするために使用されます。しかし、リアルタイム・クロックなどの一部のシステム要素の場合は、バスインターフェースだけをリセットするために **BnRES** を使用してもかまいません。このようなシステム要素は通常、別のリセット入力を持っており、これを使用して最初の電源投入時や試験目的で、コンポーネントコアをリセットすることができます。

4.8.3 転送タイプ

転送が開始する前に、バスマスターはその転送のタイプを **BTRAN[1:0]** を使用して示します。以下の転送タイプが設定できます。

- ADDRESS-ONLY
- NONSEQUENTIAL
- SEQUENTIAL

表 1-1 には、**BTRAN[1:0]** 信号のエンコーディングを示します。

表 4-1 BTRAN のエンコーディング

BTRAN [1]	BTRAN [0]	転送タイプ	説明
0	0	ADDRESS-ONLY	データの移動が不要な場合に使用します。以下に ADDRESS-ONLY 転送の3つの主用途を示します。 • IDLE サイクル • バスマスター・ハンドオーバー・サイクル • データ転送に関係しない、推測によるアドレスデコーディング
0	1	-	予備
1	0	NONSEQUENTIAL	単独転送またはバーストの最初の転送に使用します。その転送のアドレスは前のバスアクセスと無関係です。
1	1	SEQUENTIAL	バースト内の連続転送に使用します。SEQUENTIAL 転送のアドレスは常に前の転送と関係があります。

次のサイクルでデータ転送が必要かどうかを判断するために **BTRAN[1]** を使用できることが、上表から分かります。

BTRAN 信号は、**AGNTx** 入力が HIGH のときの **BCLK** HIGH フェーズ中にバスマスターによって駆動されます (図 4-13 参照)。

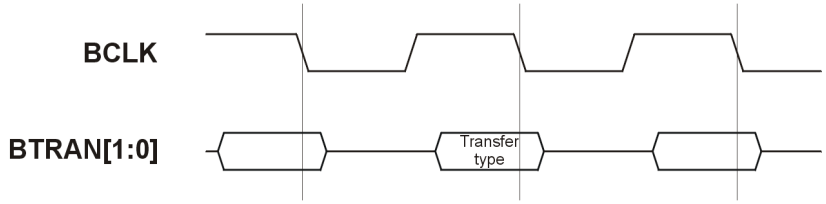


図 4-13 BTRAN タイミング

マルチマスター・システムにおいて、**BTRAN** を駆動するバスマスターは、延長された転送中に変わる場合があります。したがって、**BWAIT** LOW レベルを示し、前の転送が完了した時のみ、**BTRAN** は有効とみなされます。

4.8.4 アドレスおよび制御情報

アドレスおよび制御信号は以下の通りです。

- アドレスバス - **BA[31:0]**
- 転送方向 - **BWRITE**
- 転送サイズ - **BSIZE[1:0]**
- 保護情報 - **BPROT[1:0].**

4.8.5 アドレスバス

32 ビット・アドレスバス **BA[31:0]** は転送のアドレスを送出します。全転送がメモリマップされているので、システム内のメモリと周辺ユニットはすべて、転送がアクセスするアドレス範囲を持っている必要があります。デコーダは、どのバススレーブにアクセスすべきかを決めるためにアドレスバス (通常、高位ビット) を使用します。

4.8.6 転送方向

BWRITE 信号は、転送の方向を示すために使用されます (表 -2 参照)。**BWRITE** が LOW のとき、その転送は読出しアクセスで、HIGH のときは、書込みアクセスです。

表 4-2 BWRITE のエンコーディング

BWRITE	転送方向
0	読出し転送
1	書込み転送

4.8.7 転送サイズ

BSIZE[1:0] は転送のサイズをコード化します (表 4-3 参照)。バイト、ハーフワード、およびワードが規定され、最後のエンコーディングは今後の用途のための予備となっています。

表 4-3 BSIZE のエンコーディング

BSIZE		転送幅
[1]	[0]	
0	0	バイト (8 ビット)
0	1	ハーフワード (16 ビット)
1	0	ワード (32 ビット)
1	1	予備

バイト転送、ハーフワード転送など、データバスより狭い転送を実行する場合、バスマスターはバスの幅方向にデータを複製して、バスマスターを事実上バイエンディアンにします。読出しサイクルに応答する場合、代表的なスレーブはバス上のデータを複製しません。したがって、スレーブが駆動しているのと同じバイトレーン上にデータがあるとマスターが予想していることは重要なことです。

4.8.8 保護情報

バスマスターは BPROT 信号を使用して、実行中の転送についての追加情報を提供します (表 4-4 参照)。この情報は主に、バス保護装置として機能しているデコーダが使用するためのもので、ほとんどのバススレーブはこれらの信号を使用しません。

表 4-4 BPROT のエンコーディング

BPROT		転送特権
[1]	[0]	
-	0	オペコード取出し
-	1	データアクセス
0	-	ユーザーアクセス
1	-	特権アクセス

4.8.9 アドレスおよび制御信号のタイミング

アドレスおよび制御情報は、バスマスターが **BCLK** の立上がりエッジから生成されます。しかし、アドレスおよび制御情報のタイミングは、NONSEQUENTIAL 転送タイプと SEQUENTIAL 転送タイプに対して分けて考慮されます。これはバスマスターが通常、各々の場合においてかなり異なったタイミングパラメータを持っているからです。

図 4-14 に示すように、バスマスターが SEQUENTIAL 転送に対する高速のアドレスおよび制御出力有効タイミングを持っていることが共通の特徴です。これは、バスマスターが通常、転送開始のかなり前に SEQUENTIAL アドレスを生成できるからです。したがって、バスマスターからの出力有効時間は主に、新しい値をバス上に送るのに必要な時間に左右されます。

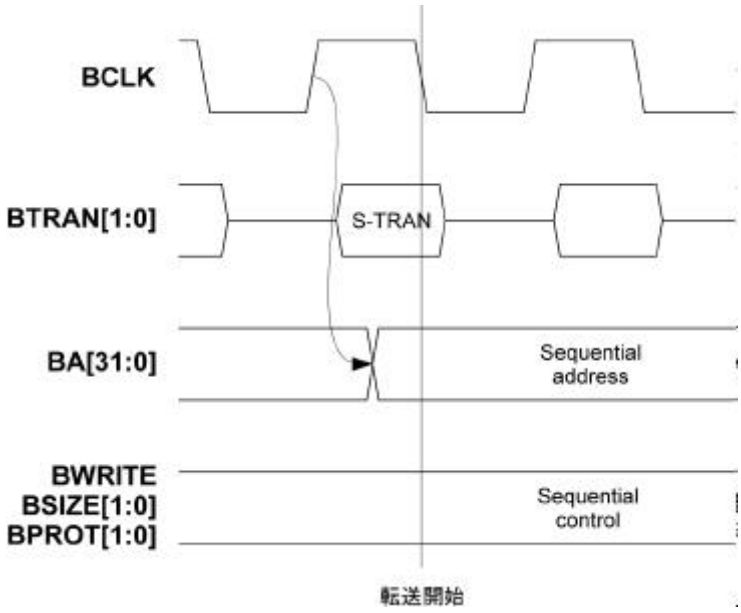


図 4-14 シーケンシャルアドレスおよび制御のタイミング

NONSEQUENTIAL の場合、SEQUENTIAL 転送の場合と比較して、バスマスターがアドレスおよび制御信号に対してかなり遅い出力有効時間を持っていることがよくあります。これを図 4-15 に示します。

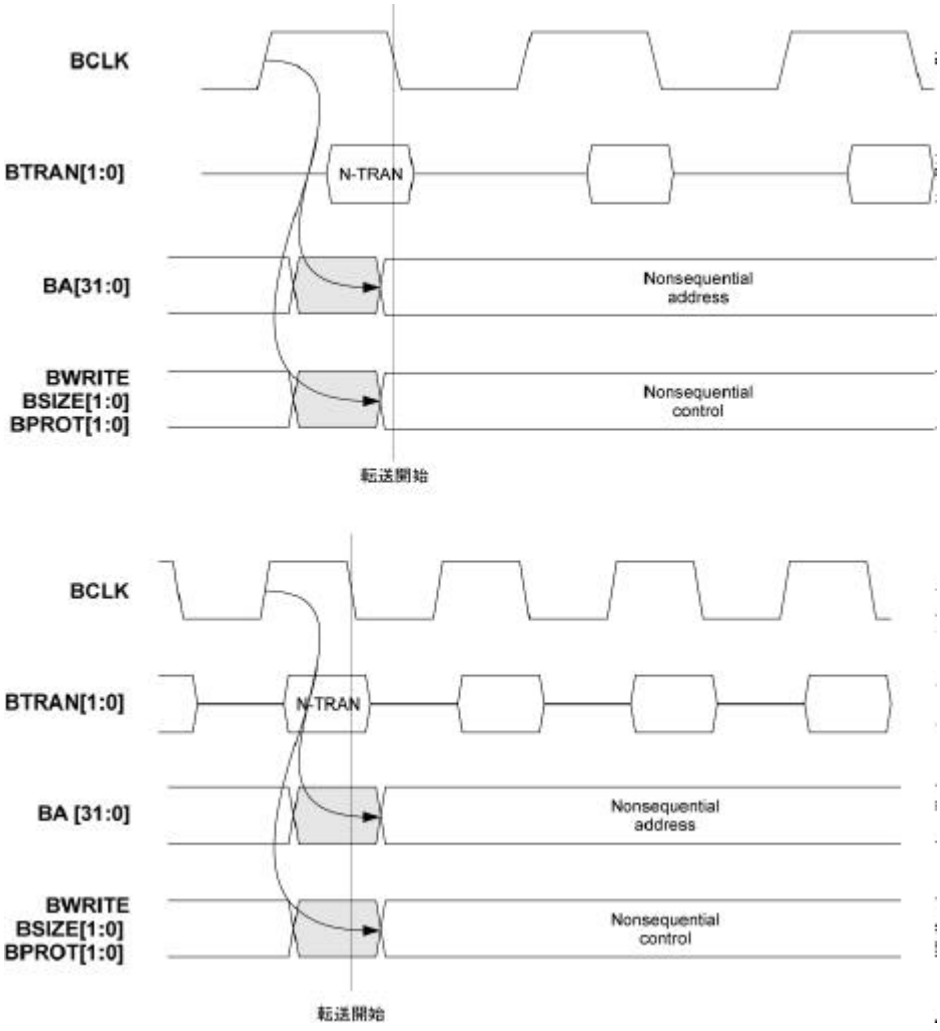


図 4-15 低周波数および高周波数クロックによるノンシーケンシャルアドレスと制御のタイミング

クロック周波数が最大値に近づいているシステムにおいて、アドレスと制御出力有効時間はクロックフェーズより大きいのが普通です。したがって、図 4-15 に示すように、アドレスは転送の開始時の **BCLK** の LOW フェーズまで有効になりません。

ADDRESS-ONLY 転送の場合、アドレスと制御情報は有効でありません。図 4-16 に示すように、ADDRESS-ONLY 転送の直後に SEQUENTIAL 転送という特殊な場合では、バスマスターは ADDRESS-ONLY 転送中にアドレスと制御情報を生成します。その結果、その情報は SEQUENTIAL 転送の開始前の **BCLK** の HIGH フェーズ間ずっと有効になります。

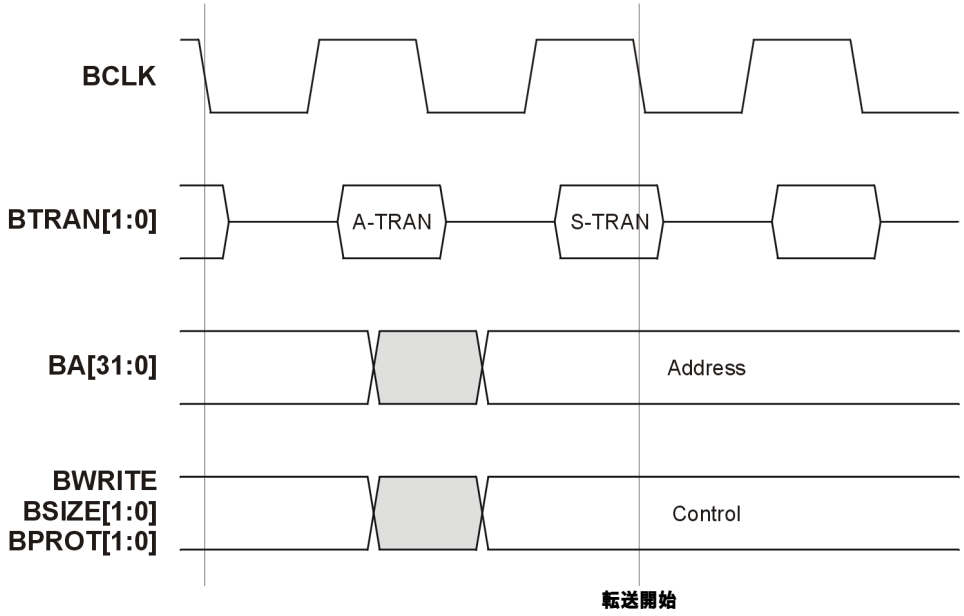


図 4-16 アドレスオンリー転送とそれに続くシーケンシャル転送 アドレスと制御のタイミング

4.8.10 アドレスおよび制御信号のトライステートイネーブル

バスマスターは、バス使用権を与えられた時、アドレスと制御信号を駆動するだけです。バス・ターンアラウンドの一期間を考慮して、バスマスターは最初にバスの使用権を与えられた時には、最初の転送前の **BCLK** の HIGH フェーズ内で駆動しません。その代わり、バスマスターは常に、ADDRESS-ONLY 転送でバス所有期間を始めます。アドレスと制御信号は、ADDRESS-ONLY 転送の **BCLK** の LOW フェーズまで駆動されません (図 4-17 参照)。

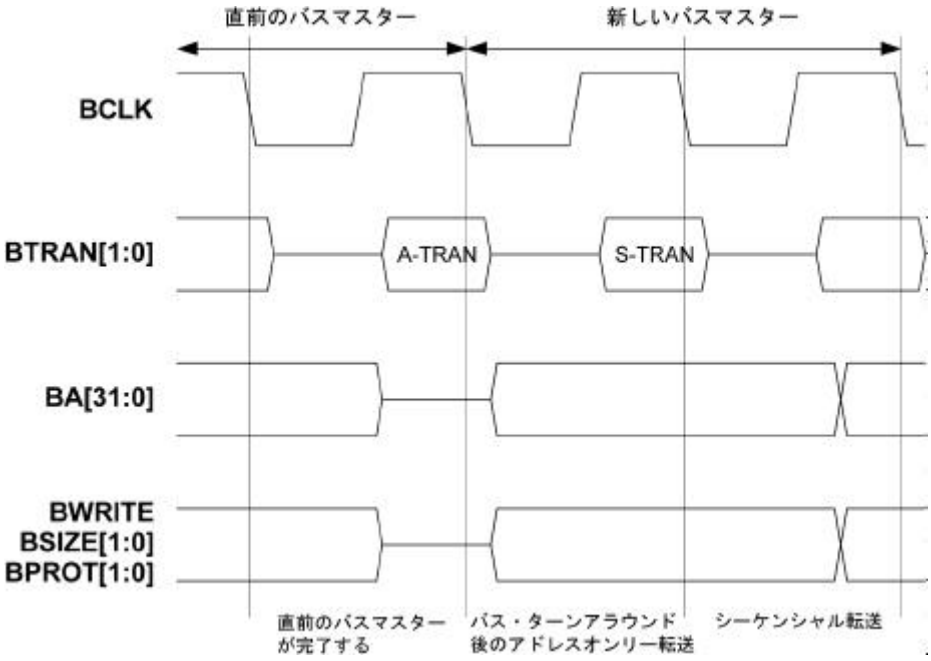


図 4-17 バスマスター・ハンドオーバー中のアドレスおよび制御信号

マルチマスターシステムの場合

- マスターが初めて認可されたとき、それらの信号は、そのマスターがバス上でアクティブになった時の **BCLK** の最初の LOW フェーズまで駆動されません。
- バスマスターがバスの使用権を与えられたとき、そのバスマスターは、**BCLK** の両方のフェーズ間、アドレスと制御信号を駆動します。
- バスマスターがバスの所有権を失ったとき、そのバスマスターは最後の **BCLK** の HIGH フェーズでそれらの信号を駆動するのを止めます。

4.8.11 スレーブ選択信号

システム内の各 ASB は **DSEL** 選択入力信号を持っています。この信号は、スレーブが転送応答を供給していること、そしてデータ転送が必要とされていることを示します。信号名 **DSELx** は、スレーブ **x** への **DSEL** 信号を示します。

ASB 上では、各スレーブには 1 つの **DSELx** 信号があります。これらの信号はデコーダで生成されます。**DSELx** 信号は転送中、ただ 1 つだけアクティブです。ADDRESS-ONLY 転送中の様に、どの **DSELx** 信号もアクティブでない時に、サイクルが行なわれます。

DSELx は、転送開始前の **BCLK** の HIGH フェーズ中に変化し、転送中は有効のままです。その信号は、**BWAIT** の LOW の転送応答に続く **BCLK** の HIGH フェーズ内に、次の転送向けに変化します。

システム設計時、ASB デコーダの実装には以下の 2 つのオプションがあります。

- *デコーダサイクルがあるデコーダ*
- *デコーダサイクルがないデコーダ*：P.4-34

この選択は設計段階では固定されており、システムのタイミング解析に基づいています。一般に、プロセッサの最高速度で動作するシステムは DECODE に数サイクルを必要とします。DECODE に数サイクルを必要としないのは、最高周波数よりかなり低い周波数で動作するシステムだけです。

デコードサイクルのあるデコーダ

高いクロック周波数を備えたシステムでは、アドレスのデコードやスレーブの選択を 1 フロックフェーズ以内で行うような危険なパスは、最大バスクロック速度を制限することになってしまいます。このようなシステムでは、デコーダを使用して、すべての NONSEQUENTIAL 転送の開始時にウェイトステート、すなわち DECODE サイクルを自動的に挿入することができます。アドレスデコーディングの危険なパスを SEQUENTIAL 転送時には回避できることが知られており、このサイクルの実施は、ウェイトステートの追加無しでの SEQUENTIAL 転送動作の継続を可能にします。従って結果的には、バスの周波数帯域の総合改善になります。

NONSEQUENTIAL 転送の場合、下図 4-18 に示すように、**DSELx** は DECODE サイクル中の **BCLK** の HIGH フェーズでアクティブになります。

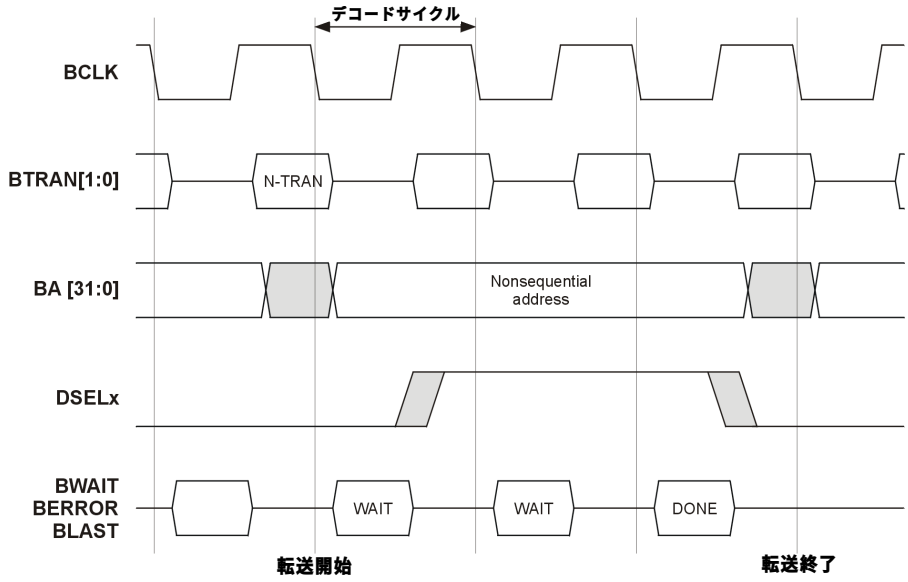


図 4-18 デコードサイクルのある選択信号のタイミング

- DECODE サイクルが実装されたとき、**DSELx** のタイミングは **BTRAN[1:0]** にのみ依存し、アドレスと制御信号には左右されません。これは、ADDRESS-ONLY 転送の場合にはどの **DSELx** 信号もアクティブにならないからです。
- NONSEQUENTIAL 転送の場合、DECODE サイクルが挿入されて、アドレスと制御情報が有効になるように 1 フェーズ全体が使用できます。
- SEQUENTIAL 転送の場合、前のサイクルからのアドレスと制御情報が使用されます。

デコードサイクルのないデコーダ

クロック周波数が低いシステムにおいて、アドレスと制御情報は、1 クロックフェーズ以内でアドレスをデコードし、スレーブを選択するのに間に合うように有効になります。このようなシステムでは、DECODE サイクルは必要ありません (図 4-19 参照)。

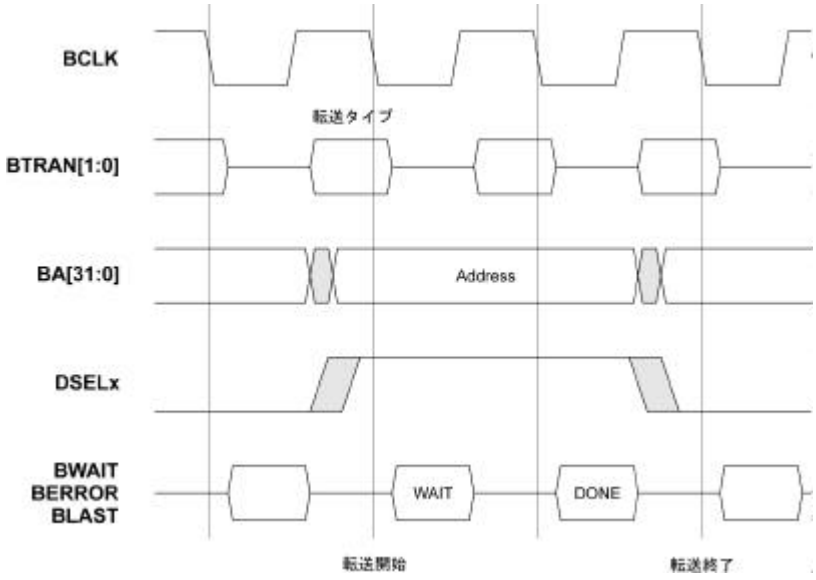


図 4-19 デコードサイクルのない選択信号のタイミング

選択信号は、転送が始まる前の **BCLK** の HIGH フェーズ時に有効になり、最後のサイクルの HIGH フェーズまで、その転送の間ずっと有効のままです。

デコーダが DECODE サイクルを挿入しない場合、**DSELx** のタイミングは、現在認可されているバスマスターが生成したアドレスと制御信号のタイミングに依存するようになります。

4.8.12 転送応答

転送状態を示すのに、スレーブデバイスは転送応答信号を使用します。

BCLK の LOW フェーズ中に、有効な転送応答が返されなければなりません。**DSELx** がアクティブになり、スレーブが選択されたときは必ず、スレーブが応答を返さなければなりません。スレーブが選択されていない場合、例えば ADDRESS-ONLY 転送中は、応答はデコーダによって返されます。

待ち応答

BWAIT は転送がいつ完了するかを示します。現在の転送を完了するためにスレーブが追加のバスサイクルを必要とするときは、**BWAIT** は HIGH になります。**BWAIT** の LOW は、その転送が終了できることを示します。転送が成功した上で完了したかどうかは、その他の転送応答信号を調べることによってのみ判断できます。

エラー応答

エラー状態は **BERROR** 信号で知らせます。これは転送失敗、すなわち、スレーブデバイスが、存在しないアドレスあるいはプロテクション・エラーが起きているアドレスへの転送になっているのを知らせるのに使用します。

多くの単純なバススレーブはエラーロジックを実装していないので、**BERROR** が LOW という固定された応答を持っています。

BERROR はまた、**BLAST** と共に使用されて、RETRACT 動作を示します。これらの信号が両者とも HIGH のとき、これはバス RETRACT が必要であることを示します。

最終応答

BLAST は、現在の転送がバーストの最後かどうかを知らせるために使用します。これは通常、ページ境界または他のバースト長さ限度を越えてバーストが続行するのを防止するために使用します。

BLAST は、次の転送がバースト転送ではなく、むしろ NONSEQUENTIAL 転送と同じ特性を持っていることを確実にするのにデコーダに使われます。これは通常、新しいアドレスデコードを実行する十分な時間があるかどうかの確認を伴います。

多くのバススレーブデバイスは、任意の数のバーストアクセスを受け入れることができ、これらのスレーブは **BLAST** が LOW という固定された応答を持っています。

BLAST はさらに、**BERROR** と共に使用されて、RETRACT 動作を示します。これらの信号が両者とも HIGH のとき、これはバス RETRACT が必要であることを示します。

バスリトラクト

ウェイトステートの少ない状態で、転送を完了することを保証できないスレーブはともすれば、バスをブロックしたり、より高い優先順位の転送が行なわれるのを阻止したりするかもしれません。このようなスレーブがシステム全体の待ち時間に影響を及ぼすのを防止するため、RETRACT メカニズムが提供されています。これによりスレーブは、転送が現時点では完了できないが、この動作は成功した上での完了まで再試行すべきであることを示します。

RETRACT は、図 4-20 に示すように、２段階で行なわれます。

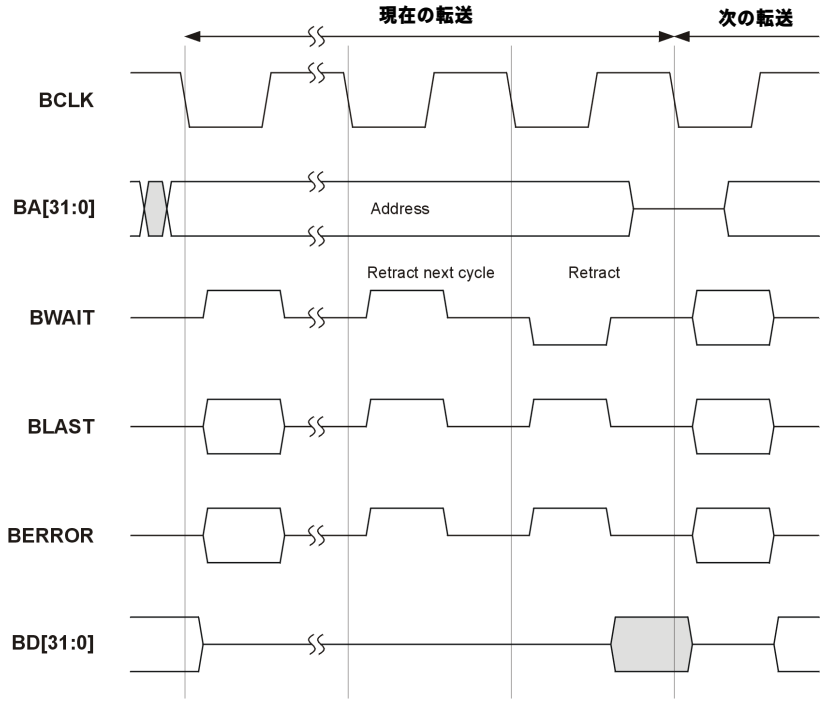


図 4-20 リトラクト動作

- 最初に、スレーブは **BWAIT**、**BLAST**、および **BERROR** をすべて HIGH にして応答します。これは RETRACT が行なわれようとしており、かつ転送が次のバスサイクルで終了することを示します。
- 2 番目のサイクルでは、転送応答は **BWAIT** が LOW で、**BLAST** と **BERROR** の両者が HIGH になります。これは転送が取り消されてしまい、かつバスが自由に使用できることを示します。

保証された完了時間を持っている基本的なスレーブは、バス RETRACT メカニズムをサポートする必要はありません。

応答の組合わせ

表 4-5 には、3 つのスレーブ転送応答信号の組合わせを示します。

表 4-5 転送応答の組合わせ

BWAIT	BLAST	BERROR	状態	説明
0	0	0	DONE	完了、転送成功
0	0	1	ERROR	完了、転送エラー
0	1	0	LAST	完了、バースト続行不能
0	1	1	RETRACT	完了、バス RETRACT
1	0	0	WAIT	未完了、待ちサイクル挿入
1	0	1	-	予備
1	1	0	-	予備
1	1	1	RETNEXT	バス RETRACT の次サイクル

バスが同期したままであること保証するため、転送応答をサイクルごとに駆動する必要があります。バス転送中は、スレーブが選択され、その **DSELx** 信号がアクティブになると、そのスレーブが転送応答信号を駆動します。

バスデコーダは、以下の動作中に転送応答を駆動します。

- ADDRESS-ONLY 転送
- DECODE サイクル
- スレーブが定義されていないアドレス空間への転送
- アクセス許可が満たされていないときの、保護領域への転送
- メモリシステムによってサポートされていない未調整転送

転送応答のタイミング

転送応答信号は、**BCLK** の立上がりエッジの前に有効に設定しなければなりません (図 4-20 参照)。

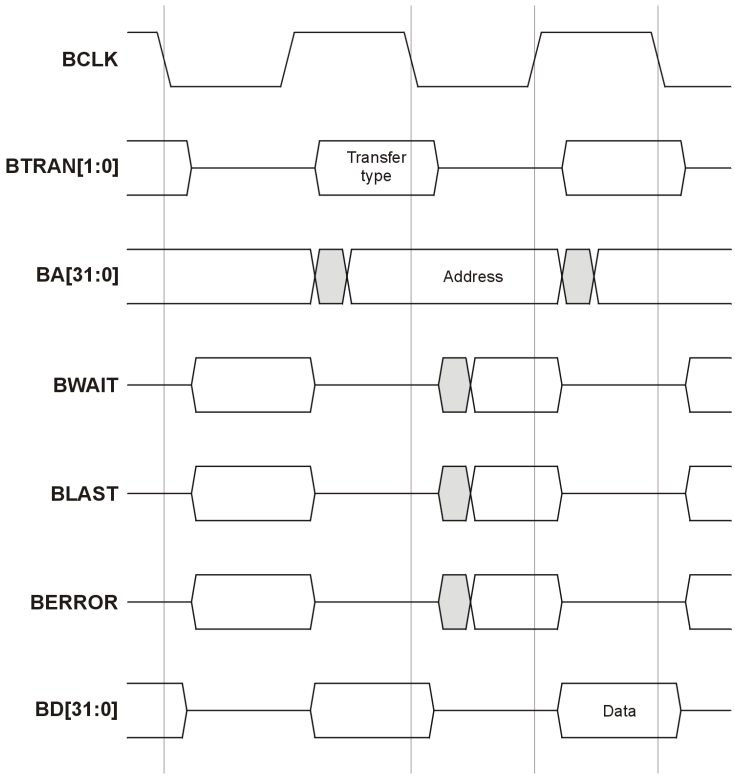


図 4-21 転送応答信号のタイミング

これらの信号は、信号ドライバ間のターンアラウンドに 1 フェーズ全体を与えるため、**BCLK** の HIGH フェーズ中には駆動されません。

4.8.13 データバス

双方向データバス **BD[31:0]** は、バスマスターとスレーブ間でデータを転送するために使用します。その転送のサイズと方向は、P.4-27「*アドレスおよび制御情報*」に記載しているように、制御信号で発行します。

データバスは、NONSEQUENTIAL 転送の最初の **BCLK** の LOW フェーズ中に駆動してはいけません。このバスはリセット時以外は適切なマスターまたはスレーブによって駆動します。

書込み転送時

- NONSEQUENTIAL 転送の最初の **BCLK** の LOW フェーズを除く、転送の全フェーズ中は、マスターがデータバスを駆動します。
- スレーブはデータバスを駆動しません。

読出し転送時

- マスターはデータバスを駆動しません。
- 転送の最後の **BCLK** の HIGH フェーズ中は、スレーブがデータバスを駆動しなければなりません。転送の残りのフェーズの場合、スレーブがデータバスを駆動してもよいし、トライステート状態にしておいてもかまいません。ただし、NONSEQUENTIAL 転送の最初の **BCLK** の LOW フェーズ中は、データバスは駆動されません。

以下のダイアグラムでは、データバスの駆動方法についていくつかの例を示します。

図 4-22 には、NONSEQUENTIAL 書込み転送の例を示します。

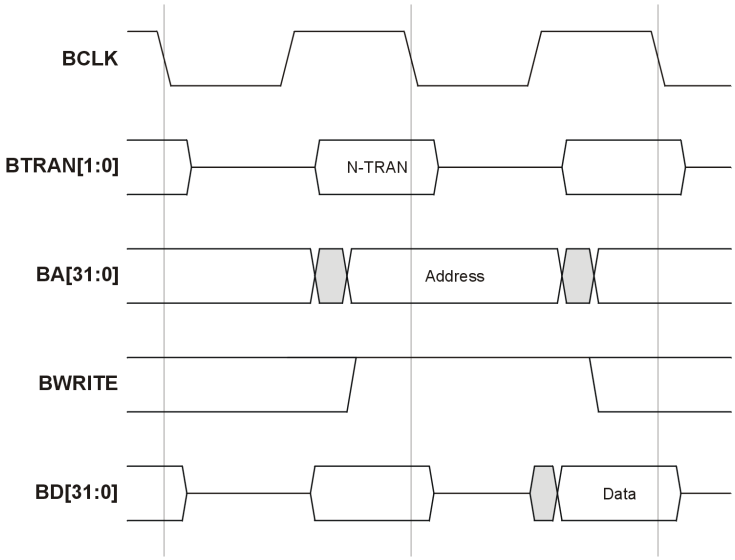


図 4-22 ノンシーケンシャル書込み転送

データバスは、最初のサイクルの **BCLK** の LOW フェーズ中を除き、バスマスターに駆動されます。NONSEQUENTIAL 転送の開始時にデータバスを駆動しないことにより、異なったデータバス・ドライバ間のターンアラウンドのために 1 フェーズ全体が得られます。

BWAIT を使用して書込み転送を延長した場合、図 4-23 に示すように、その転送を完了するために必要な追加サイクルの **BCLK** の LOW フェーズ中、データは有効のままです。

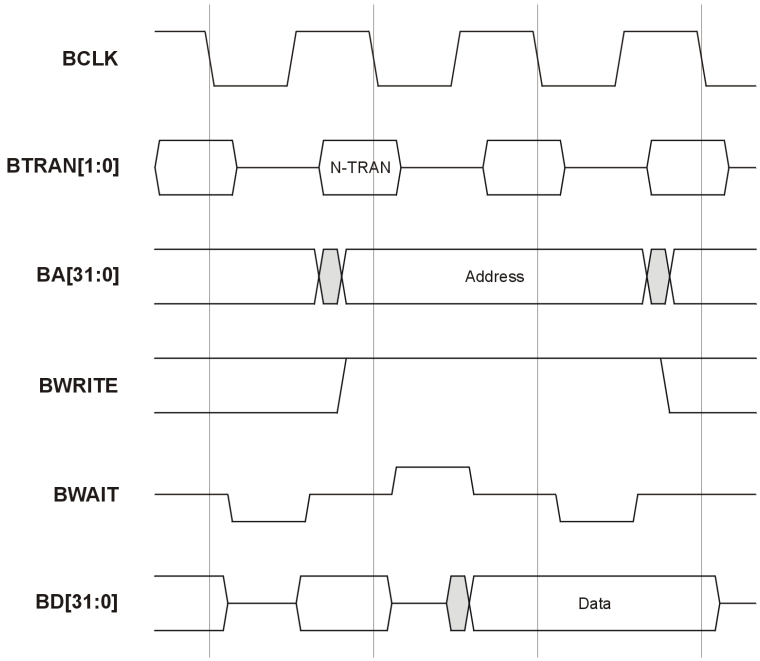


図 4-23 延長された書込み転送

SEQUENTIAL 転送の場合、図 4-24 に示すように、その転送開始時の **BCLK** の LOW フェーズ中、バスマスターがデータを駆動します。これは、SEQUENTIAL 転送には 1 フェーズのターンアラウンドが必要ないので、許可されます。

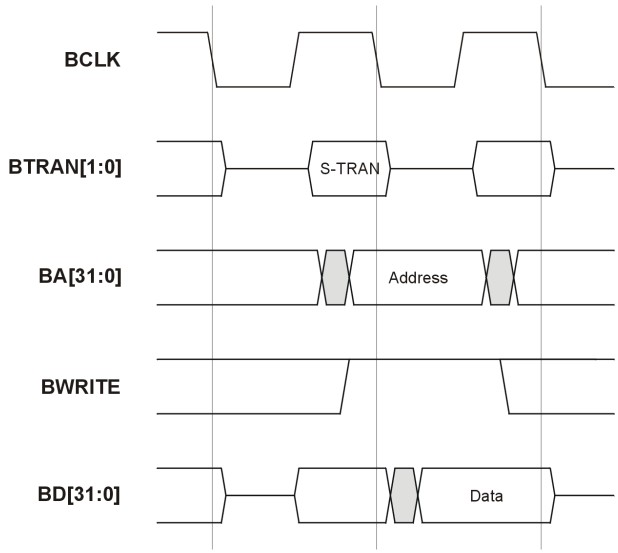


図 4-24 シーケンシャル書込み転送

読出しサイクル中は、スレーブがデータバスを駆動します。書込みサイクルの場合のように、NONSEQUENTIAL 転送の場合、最初のサイクルの **BCLK** の LOW フェーズではデータバスは駆動されません (図 4-25 参照)。転送の残りのフェーズではずっと、バススレーブがバスを駆動します。

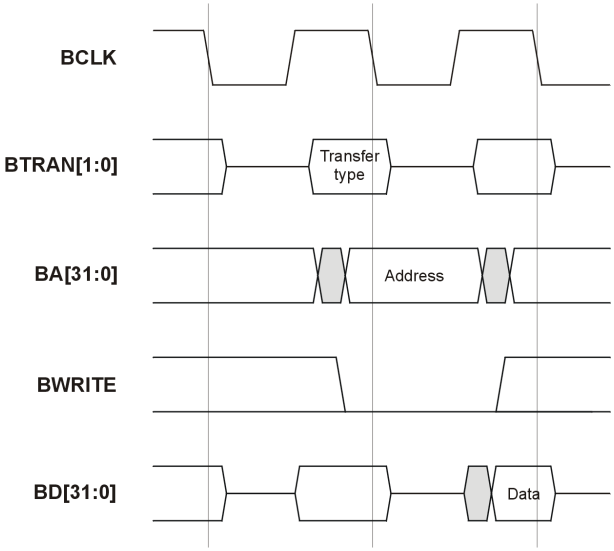


図 4-25 読出し転送

転送の間中、スレーブがデータバスを駆動する必要はありません。唯一の必要条件は、転送の最終 **BCLK** の HIGH フェーズの終了付近で有効であるようにデータを駆動することです。

4.8.14 アービトレーション信号

AREQx - バス要求

AREQx は、マスターからアービタへの要求信号で、そのマスターがバスを必要としていることを示します。各マスターが **AREQx** 信号を持っており、この信号は **BCLK** の HIGH フェーズ中に変化します。

AGNTx - バス認可

アービタからバスマスターへの認可信号は、そのバスマスターが現在、バスを要求しているマスターの中で最も優先順位が高いことを示します。システム内の各バスマスターに **AGNTx** 信号があります。

AGNTx は、どのマスターが現在バスの使用権を得ているかを示している訳ではないことに注意が必要です。その代わりに現在、最優先順位であるマスターを示します。**BWAIT** が LOW になり、転送が完了した時点では、アクティブな **AGNTx** を持っているマスターがバスの使用を認可されます。

AGNTx は、**BCLK** の LOW フェーズ中に、アービタによって変えられますが、HIGH フェーズ中はずっと有効のままです。

AGNTx が HIGH のとき、該当するマスターは以下の動作を行ないます。

- **BCLK** が HIGH 中に **BTRAN** 信号を駆動します。
- **BWAIT** が LOW のとき、認可されます。

BLOK - バスロック

BLOK は共用バスロック信号です。この信号は、次の転送が現在の転送と分割不可であり、他のバスマスターにバスへのアクセス権を与えるべきでないことを示します。

マスターは、たとえそのマスターが転送を行なっていない場合でも、バスの使用を認可されたときには必ず、**BLOK** 信号を有効レベルにしなければなりません。これはアービトレーションプロセスが続行できるようにするために必要です。

BLOK が LOW のとき、アービタは、バスを要求しているバスマスターのうち最優先のマスターに認可します。

BLOK が HIGH のとき、アービタは同じマスターを認可状態のままにしておきます。

BLOK は、**BCLK** の LOW フェーズ中に、アービタに取得されます。この信号は有効でなければならず、その結果、アービタは **BCLK** の立上がりエッジの前に有効な **AGNTx** 出力を生成できます。**BLOK** は、バスマスター・ハンドオーバーサイクル中はアービタから無視されます。

4.9 ASB AMBA コンポーネントについて

本セクションでは、AMBA システム内の各要素について説明し、ASB ベース AMBA 設計の解析に必要な一般的なタイミングパラメータについて説明する。

タイミングパラメータに対して以下の表記法を使用します。

- T_{is} - 入力設定時間
- T_{ih} - 入力ホールド時間
- T_{ov} - 出力有効時間
- T_{oh} - 出力ホールド時間

別に記載がない限り、タイミングパラメータは信号の立上がりエッジと立下がりエッジの両方に適用されます。トライステートイネーブルおよびディセーブル時間は、明確に規定されていません。すべてのトライステートディセーブル時間は、バスの衝突を防止するために BCLK の 1 フェーズより短くなければなりません。トライステートドライバのイネーブル化が支配的な要因であるならば、場合によっては、トライステートイネーブル時間を出力有効時間の要因として織り込む必要があります。

4.10 ASB バススレーブ

ASB バススレーブはシステム内のバスマスターが起動する転送に応答します。本スレーブはデコーダからの DSEL 選択信号を使用して、いつバス転送に応答すべきかを判断します。アドレスと制御情報など、転送に必要な他の信号はすべて、バスマスターが生成します。

デコーダは、スレーブインターフェースを非常に単純化し、バス上で行なわれる可能性のある様々なタイプの転送についてスレーブが理解する必要性を無くします。

4.10.1 インターフェースダイアグラム

図 4-26 には、ASB バススレーブ・インターフェースを示します。

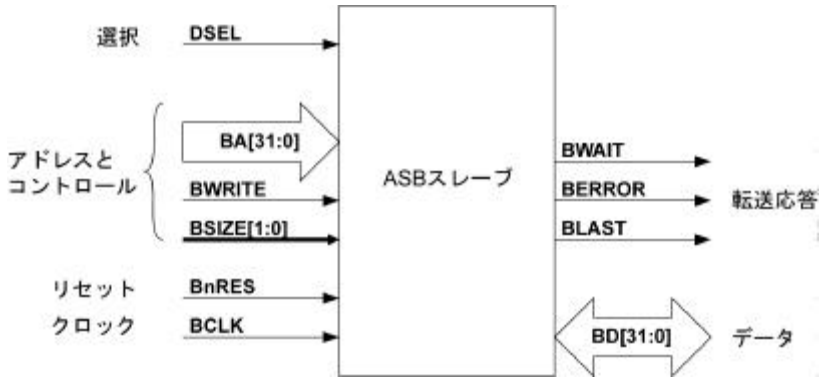


図 4-26 ASB バススレーブ・インターフェース

4.10.2 バススレーブ・インターフェースの説明

バススレーブ・インターフェースを以下の項目について説明します。

- *転送応答*
- *データ*：P.4-48

転送応答

スレーブは、DSEL がアクティブになっているときの BCLK の LOW フェーズで転送応答を返します。BWAIT、BERROR、および BLAST 信号を使用して、以下の応答のうちの 1 つを生成しなければなりません。

WAIT　　転送が完了する前に、その転送を延長しなければなりません。

DONE	転送が成功した上で完了しました。
LAST	転送は成功した上で完了したが、スレーブがこれ以上のバースト転送を受け入れることができないか、あるいはメモリ境界に到達した。
ERROR	転送は成功した上で完了しませんでした。このエラー状態はバスマスターに信号で通知されます。その結果、バスマスターはその転送は成功しなかったことを知ります。
RETRACT	転送がまだ完了していません。それで、バスマスターは転送を再試行すべきです。完了に多くのサイクルを要する転送によってバスがロックされるのを防ぐために、スレーブは RETRACT 応答を使います。

多くのスレーブは WAIT や DONE 応答だけを使用します。この場合、転送応答が発行されると、**BERROR** と **BLAST** の両方が LOW になります。

DSEL が LOW を示し、スレーブが選択されていない場合、転送応答信号はトライステートです。応答信号はリセット中もトライステートでなければなりません。

データ

スレーブインターフェースは単一ステートマシンとして実装され、クロックの立下がりエッジから動作して、いつデータ転送を行えるかを判断します。リセット中は、ステートマシンは NOT_SELECTED 状態に入ります (図 4-27 参照)。

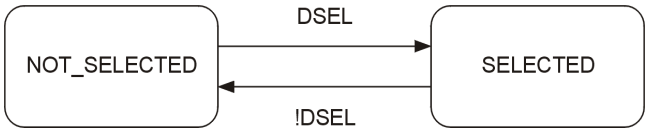


図 4-27 ASB スレーブのバスインターフェース・ステートマシン

書込み転送では、スレーブが SELECTED 状態にあるとき、クロックの立下がりエッジでデータを取得します。また、必要に応じて、上述の転送応答信号を使用して、転送を延長する場合があります。

読出し転送では、スレーブは転送の最終クロックの HIGH フェーズ中、データバスを駆動しなければなりません。

待機用サイクルの挿入により転送が延長された場合、スレーブは転送の追加サイクル中、データバスを駆動するか、あるいは転送の最終フェーズまでデータバスをトライステート状態のままにしておきます。

転送が SEQUENTIAL であるか、それとも NONSEQUENTIAL であるかをスレーブが判断する必要を無くすためには、転送の最初のフェーズ中にデータバスを駆動しないスレーブを設計することが通常、より簡単な方法です。

リセット中、またはスレーブが NOT_SELECTED のとき、データバスはトライステートでなければなりません。

4.10.3 タイミングダイアグラム

図 4-28 には、ASB バススレーブへのアクセスと関係があるタイミングパラメータを示します。

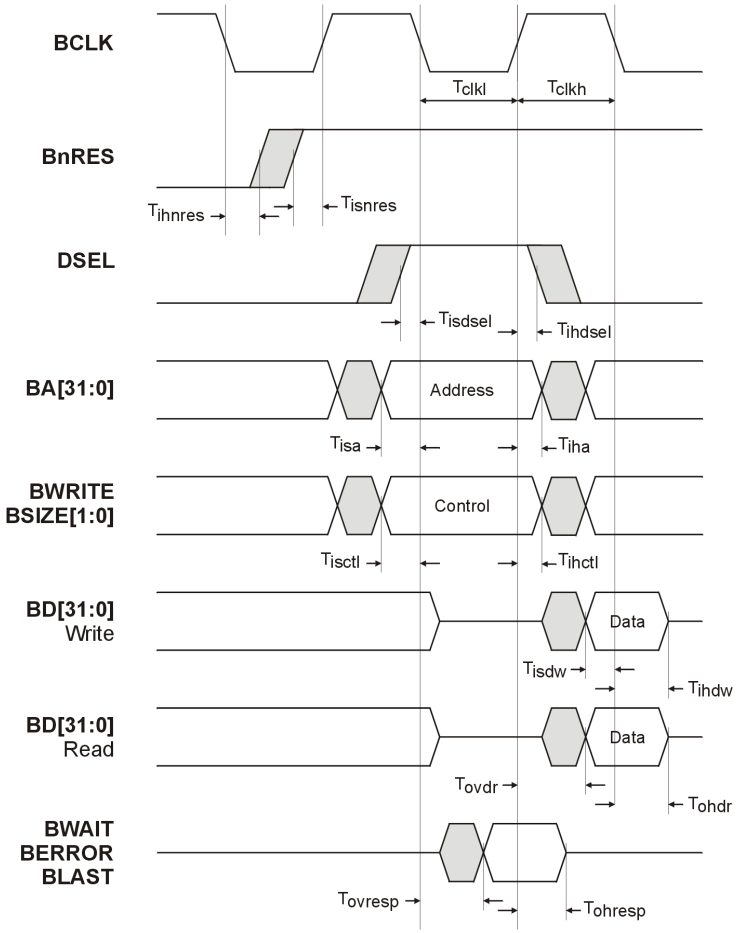


図 4-28 ASB スレーブ転送

4.10.4 タイミングパラメータ

ASB バススレーブ関連のタイミングパラメータを、入力信号について表 4-6 に、出力信号について表 4-7 に示します。双方向信号については、両方の表を参照して下さい。

表 4-6 ASB スレーブ入力パラメータ

パラメータ	説明
T_{clkl}	**BCLK** LOW **時間**
T_{clkh}	**BCLK** HIGH **時間**
T_{isnres}	**BCLK の立上がりエッジまでの BnRES インアクティブ設定時間**
T_{ihnres}	**BCLK の立下がりエッジ後の BnRES インアクティブホールド時間**
T_{isdsel}	**BCLK の立下がりエッジまでの DSEL 設定時間**
T_{ihdsel}	**BCLK の立上がりエッジ後の DSEL ホールド時間**
T_{isa}	**BCLK の立下がりエッジまでの BA[31:0] 設定時間**
T_{iha}	**BCLK の立上がりエッジ後の BA[31:0] ホールド時間**
T_{isctl}	**BCLK の立下がりエッジまでの BWRITE および BSIZE[1:0] 設定時間**
T_{ihctl}	**BCLK の立上がりエッジ後の BWRITE および BSIZE[1:0] ホールド時間**
T_{isdw}	**書込み転送の場合、BCLK の立下がりエッジまでの BD[31:0] 設定時間**
T_{ihdw}	**書込み転送の場合、BCLK の立下がりエッジ後の BD[31:0] ホールド時間**

表 4-7 ASB スレーブ出力パラメータ

パラメータ	説明
T_{ovresp}	**BCLK の立下がりエッジ後の BWAIT、BERROR および BLAST 有効時間**
T_{ohresp}	**BCLK の立上がりエッジ後の BWAIT、BERROR および BLAST ホールド時間**
T_{ovdr}	**読出し転送の場合、BCLK の立上がりエッジ後の BD[31:0] 有効時間**
T_{ohdr}	**読出し転送の場合、BCLK の立下がりエッジ後の BD[31:0] ホールド時間**

Note

BCLK の立下がりエッジでデコーダ、アドレス、および制御信号をすべて取得するようにバススレーブを設計した場合、1 フェーズ全体の入力保持時間がバスプロトコルによって保証されます。

追加のウェイトステートを転送に挿入することにより、1 フェーズ全体のホールド時間がデータバスに提供されるのを確実にします。

4.11 ASB バスマスター

ASB バスマスターは、AMBA システム内で最も複雑なバスインターフェースを持っています。通常、AMBA システム設計者は既に設計済みのバスマスターを使用します。そのためバスマスター・インターフェースの詳細を気にする必要はありません。

バスマスター・インターフェースは、テストしたりバスマスター動作のプログラミングのためのスレーブインターフェースを組み込むことも出来ます。そのような場合、多くのインターフェース信号がマスターインターフェースとスレーブインターフェース間に割り当てられます。

4.11.1 インターフェースダイアグラム

ASB バスマスターのインターフェースダイアグラムで、主な信号群を示します。

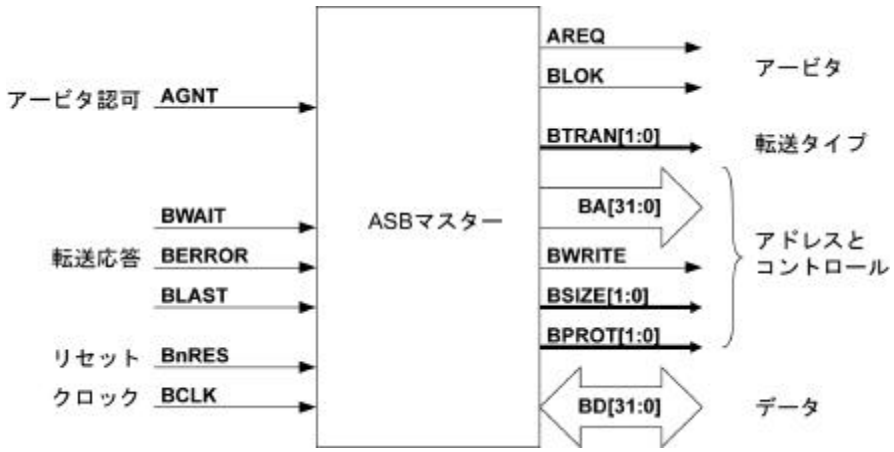


図 4-29 ASB バスマスター・インターフェース・ダイアグラム

4.11.2 バスマスター・インターフェースの説明

バスマスター・インターフェースは 2 つのステートマシンから構成されています。

- 第一ステートマシンは、マスターにバスの使用権を、現在与えているかどうかを判断します。
- より複雑な第二ステートマシンは、マスターのバスインターフェースを制御するために使用されます。

GRANTED ステートマシン

GRANTED ステートマシンは、バスマスターにバスの使用権を与えているかどうかを判断するために使用されます。それは **BCLK** の立上がりエッジに同期し、そして 2 つの状態だけ、すなわち GRANTED と NOT_GRANTED をとります。図 4-30 には、状態図を示します。

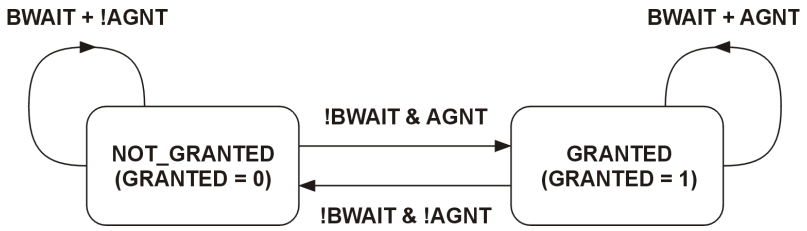


図 4-30 バスマスターの認可ステートマシン

ステートマシンからの出力は、バスマスターのメイン・ステートマシンで使用される GRANTED 信号です。

Note

AGNT 信号は、多くのクロックサイクルに対してアクティブになりますが、バスマスターが実際に GRANTED になるのは、**AGNT** がアクティブで、**BWAIT** が LOW のときのみです。

設計において、ステートマシンが **AGNT** 信号の値に左右される状態で非同期にリセットされることがあるのを考慮するのは重要な事柄です。リセット中に、システム内の 1 つのバスマスターがデファルト・バスマスターとして設定され、リセット中の **AGNT** のアクティブ変化で指し示されます。そして GRANTED 状態にリセットされます。他のすべてのバスマスターは、NOT_GRANTED 状態にリセットされます。

4.11.3 バスインターフェース・ステートマシン

バスインターフェースのメイン・ステートマシンは、立下がりエッジでトリガされ、6 つの状態を持っています。図 4-32 に示す状態図全体はかなり複雑ですが、図 4-31 に示すように 4 つの象限で考えることができます。

図 4-31 バスインターフェース・ステートマシンの象限

TRANSFER REQUEST GRANTED 象限は、バス・ターンアラウンドと RETRACT 動作を処理する 3 つの状態を持ちます。

2 つの内部バスマスター信号 GRANTED と REQUEST は、状態図中にある遷移のほとんどを制御します (図 4-32 参照)。

- GRANTED は、上述のより単純なステートマシンから生成されます。
- REQUEST は、バスマスターが直接生成します。

REQUEST は、バスマスターがバスでの転送を必要とするとき HIGH になり、バスマスターがバスへのアクセス権を必要としないとき LOW になります。

状態図中の遷移が GRANTED と REQUEST によって制御されない唯一の時間は、バスマスターが ACTIVE 状態のときです。この状態では、次の状態への遷移は、受信する転送応答によって決まります。そのダイアグラムに示されている WAIT、DONE、LAST、ERROR、および RETNEXT は、転送応答信号のエンコーディングに対応します。

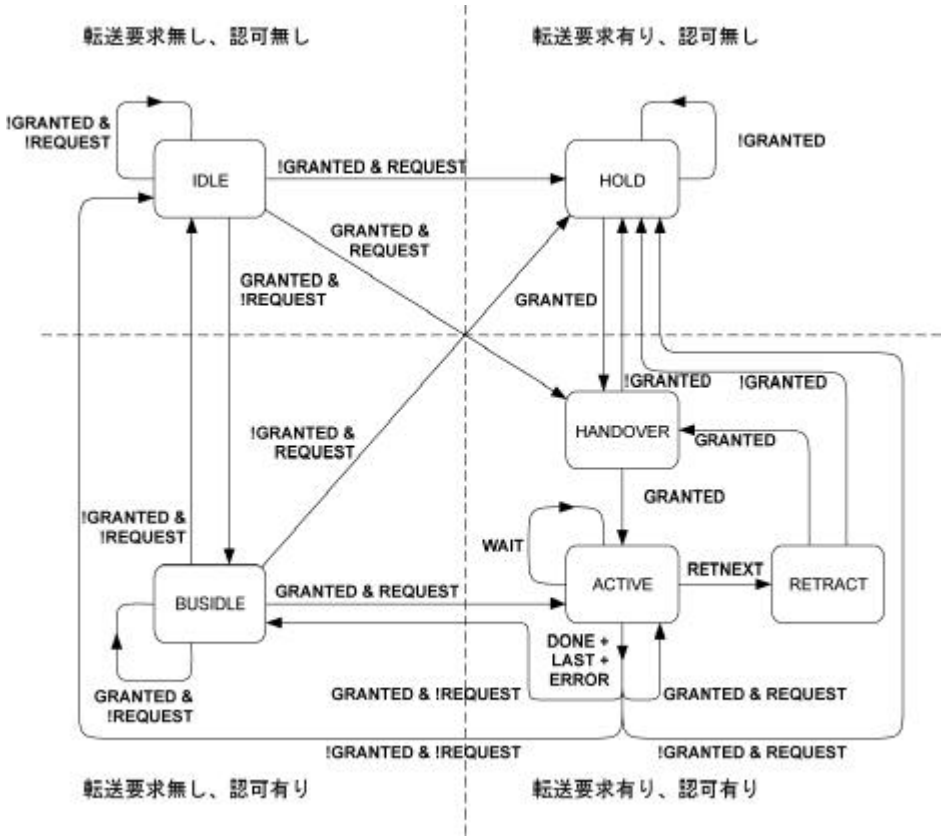


図 4-32 バスマスターのメイン・ステートマシン

状態図は次の仮定に基づいていることに注意して下さい。バスマスターが一旦 REQUEST が示すとおりに転送要求を行なうと、REQUEST は、バスマスターが転送を実行するまでアクティブのままです。

バスマスターのメイン・ステートマシンはクロックの立下がりエッジから動作しているので、ACTIVE 状態の終了を制御するのに転送応答信号 **BWAIT**、**BERROR**、および **BLAST** のラッチ付き回路を使用する必要があります。

状態図には、リセット状態が示されていません。また、認可されたステートマシンと同様に、バスマスターのメイン・ステートマシンも複雑なリセット状態を持っています。リセット中に、**AGNT** がアクティブになった場合、**BnRES** が LOW のときに、バスマスターはデフォルト・バスマスターとなり、BUSIDLE 状態に入ります。しかし、**AGNT** がリセット中にアクティブでない場合、バスマスターは IDLE 状態に入ります。

表 4-8 には、各状態において行なわれる動作を示します。

表 4-8 各状態で行なわれる動作

名前	説明	動作
IDLE	マスターはバスを必要とせず、また認可もされません。	内部 **BTRAN** は ADDRESS-ONLY。 マスタークロックは有効。 マスターのアドレスバスはトライステート。 マスターデータバスはトライステート。
BUSIDLE	マスターはバスを必要としていないのに、認可されました。	内部 **BTRAN** はマスターが示す通り。 マスタークロックは有効。 マスターのアドレスバス・イネーブルは GRANTED 信号から生成。 マスターデータバスはトライステート。
HOLD	マスターはバスを必要としましたが、認可されませんでした。	内部 **BTRAN** は ADDRESS-ONLY。 マスタークロックは無効。 マスターのアドレスバスはトライステート。 マスターデータバスはトライステート。
HANDOVER	異なるバスマスター間で切り換わると、この状態はバス・ターンアラウンドを提供します。	内部 **BTRAN** は SEQUENTIAL。 マスタークロックは無効。 マスターのアドレスバス・イネーブルは GRANTED 信号から生成。 マスターデータバスはトライステート。
ACTIVE	データ転送が行なわれるアクティブな状態。 この状態の終了は転送応答に依存します。	内部 **BTRAN** はマスターが示す通り。 マスターのクロックイネーブルは **BWAIT** から起動。 マスターのアドレスバス・イネーブルは GRANTED 信号から生成。 書込みトランザクションの場合、マスターのデータバス・イネーブルは有効。
RETRACT	取り消し状態、システム内の残りの素子は転送の終了を知ったが、バスマスターは変わらず。	内部 **BTRAN** は ADDRESS-ONLY。 マスタークロックは無効。 マスターのアドレスバス・イネーブルは GRANTED 信号から生成。 書込みトランザクションの場合、マスターのデータバス・イネーブルは有効。

BTRAN[1:0] トライステートドライバは、**AGNT** と **BCLK** の両方が HIGH の時有効になります。

マスターのアドレスバス・イネーブルは、**BA[31:0]**、**BWRITE**、**BSIZE[1:0]**、**BPROT[1:0]** および **BLOK** のトライステートイネーブルを制御するのに使用します。マスターのデータバス・イネーブルは、**BD[31:0]** のトライステートイネーブルを制御するために使用します。

4.11.4 バスマスターのタイミングダイアグラム

以下のダイアグラムは、AMBA システムで動作している ASB バスマスターに関係するタイミングパラメータを示しています。

- 図 4-33 では、*「ASB バスマスターのノンシーケンシャル転送」*を示します。
- 図 4-34 では、P.4-58 *「ASB バスマスターのシーケンシャル転送」*を示します。
- 図 4-35 では、P.4-59 *「ASB マスターのアドレスオンリー転送」*を示します。
- 図 4-36 では、P.4-60 *「ASB バスマスターのアービトレーションおよびリセット信号」*を示します。

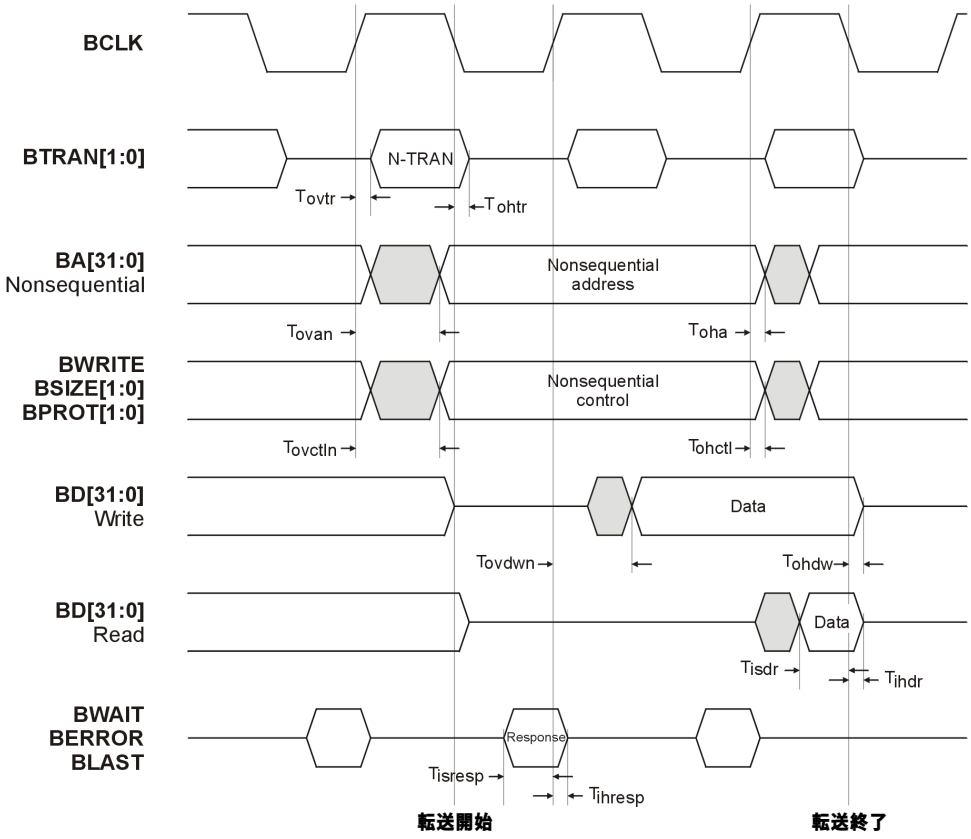


図 4-33 ASB バスマスターのノンシーケンシャル転送

図 4-33 に示す NONSEQUENTIAL 転送の場合、転送開始前の **BCLK** の HIGH フェーズでアドレスと制御信号が有効になります。AMBA プロトコルの重要な機能とは、NONSEQUENTIAL 転送に不足した出力有効時間を考慮に入れていることで、この機能は、デコーダによる全ての NONSEQUENTIAL 転送開始時における、ウェイトステートの自動挿入によって実現されます。

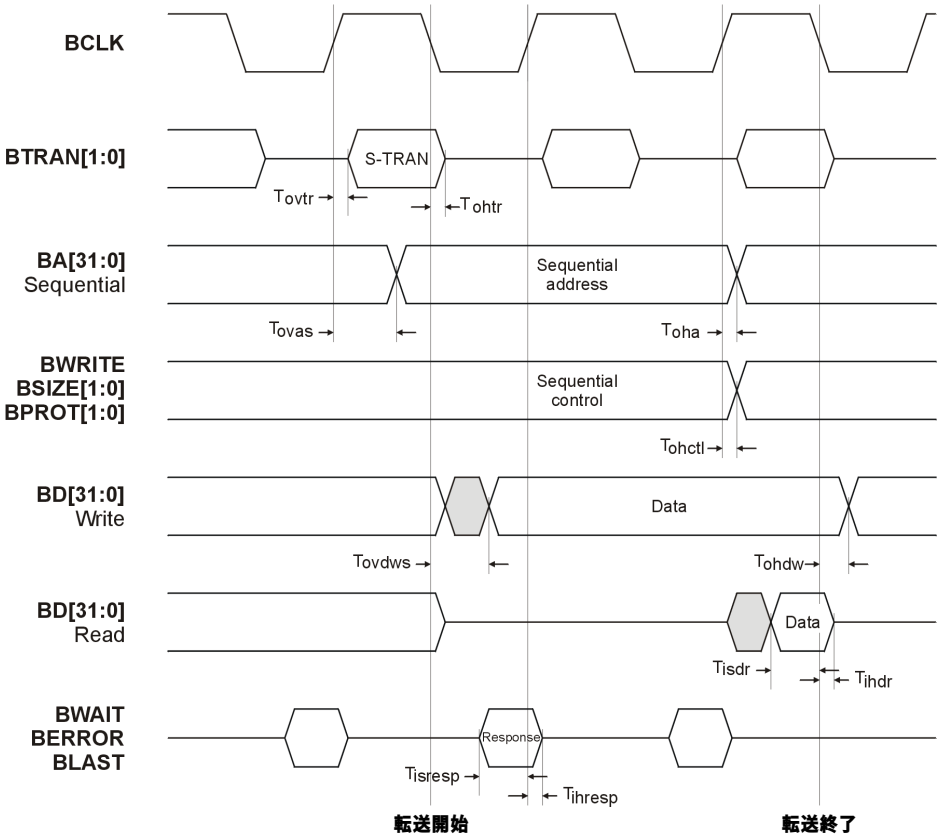


図 4-34 ASB バスマスターのシーケンシャル転送

SEQUENTIAL 転送は、アドレスと制御信号有効時間について異なったタイミングパラメータを持っています (図 4-34 参照)。代表的なバスマスターでは、SEQUENTIAL 転送の出力有効時間は、NONSEQUENTIAL 転送の場合よりはるかに好都合です。アドレス、制御、およびデータの出力ホールド時間は等しく、かつ転送タイプに依存しません。

SEQUENTIAL と NONSEQUENTIAL 転送間のもう一つの相違点は、SEQUENTIAL 転送中において、最初の転送フェーズ内でデータが駆動されてしまうことがあるので、データ有効のパラメータが **BCLK** の立下がりエッジから明記されている点です。

ADDRESS-ONLY 転送の場合、アドレスおよび制御信号は、その転送開始前のクロック HIGH フェーズ中に駆動され、またはバスマスター・ハンドオーバーの場合は、その転送自体のクロック LOW フェーズ内でのみ駆動されます (図 4-35 参照)。アドレスおよび制御有効タイミングパラメータは、ADDRESS-ONLY 転送の直後に SEQUENTIAL 転送が行なわれるときのみ関係します。この場合、アドレスおよび制御信号は、ADDRESS-ONLY 転送の LOW フェーズ中に有効になるように駆動されなければなりません。これは、それらの信号が SEQUENTIAL 転送以前のクロック HIGH フェーズの間ずっと有効になることを意味します。

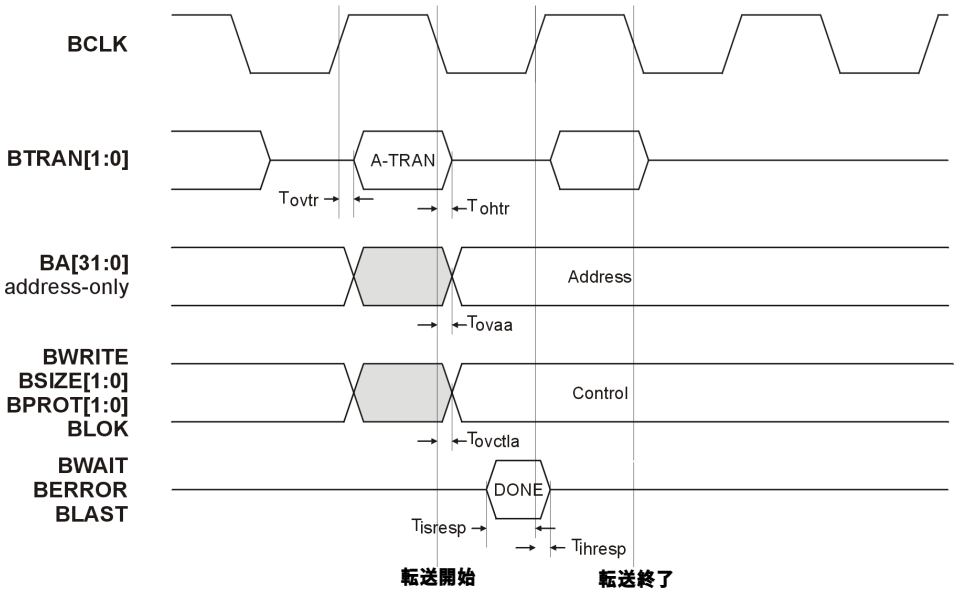


図 4-35 ASB マスターのアドレスオンリー転送

図 4-36 には、ASB バスマスターのアービトレーションとリセット信号を示します。

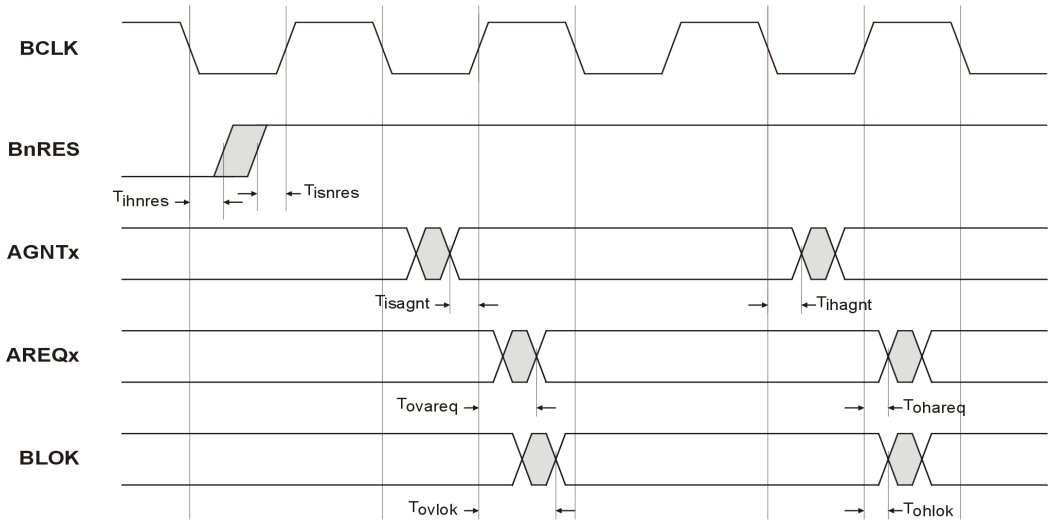


図 4-36 ASB バスマスターのアービトレーションとリセット信号

BnRES は、非同期的にアクティブになる場合があります。したがって、この信号のアクティブ化に関係する設定やホールドのパラメータはありません。バスマスターからの出力である AREQ 信号は、HIGH **クロックフェーズ中に変化し、アービタから返ってくる AGNT 信号は** LOW **クロックフェーズ中に変化します。**

4.11.5 タイミングパラメータ

AMBA システムで動作している ASB バスマスターと関係するタイミングパラメーターを以下の 2 つの表に示します。表 4-9 には入力信号について、表 4-10 には出力信号について列挙します。双方向信号については、両方の表に説明します。

表 4-9 バスマスターの入力タイミングパラメータ

パラメータ	説明
T_{clkl}	**BCLK** LOW **時間**
T_{clkh}	**BCLK** HIGH **時間**
T_{isnres}	**BCLK の立上がりエッジまでの BnRES インアクティブ設定時間**

表 4-9 バスマスターの入力タイミングパラメータ（続き）

パラメータ	説明
T_{ihnres}	**BCLK の立下がりエッジ後の BnRES インアクティブホールド時間**
T_{isresp}	**BCLK の立上がりエッジまでの BWAIT、BERROR、BLAST 設定時間**
T_{ihresp}	**BCLK の立上がりエッジ後の BWAIT、BERROR、BLAST ホールド時間**
T_{isdr}	**読出し転送の場合、BCLK の立下がりエッジまでの BD[31:0] 設定時間**
T_{ihdr}	**読出し転送の場合、BCLK の立下がりエッジ後の BD[31:0] ホールド時間**
T_{isagnt}	**BCLK の立上がりエッジまでの AGNT 設定時間**
T_{ihagnt}	**BCLK の立下がりエッジ後の AGNT ホールド時間**

表 4-10 バスマスターの出力タイミングパラメータ

パラメータ	説明
T_{ovtr}	**BCLK の立上がりエッジ後の BTRAN 有効時間**
T_{ohtr}	**BCLK の立下がりエッジ後の BTRAN ホールド時間**
T_{ovan}	NONSEQUENTIAL **転送の場合、BCLK の立上がりエッジ後の BA[31:0] 有効時間**
T_{ovas}	SEQUENTIAL **転送の場合、BCLK の立上がりエッジ後の BA[31:0] 有効時間**
T_{ovaa}	ADDRESS-ONLY **の場合、BCLK の立下がりエッジ後の BA[31:0] 有効時間**
T_{oha}	**BCLK の立上がりエッジ後の BA[31:0] ホールド時間**
T_{ovctln}	NONSEQUENTIAL **転送の場合、BCLK の立上がりエッジ後の BWRITE、BSIZE[1:0]、BPROT[1:0] 有効時間**
T_{ovctla}	ADDRESS-ONLY **転送の場合、BCLK の立下がりエッジ後の BWRITE、BSIZE[1:0]、BPROT[1:0] 有効時間**
T_{ohctl}	**BCLK の立上がりエッジ後の BWRITE、BSIZE[1:0]、BPROT[1:0] ホールド時間**
T_{ovdwn}	NONSEQUENTIAL **書込み転送の場合、BCLK の立上がりエッジ後の BD[31:0] 有効時間**
T_{ovdws}	SEQUENTIAL **書込み転送の場合、BCLK の立下がりエッジ後の BD[31:0] 有効時間**
T_{ohdw}	**書込み転送の場合、BCLK の立下がりエッジ後の BD[31:0] ホールド時間**

表 4-10 バスマスターの出力タイミングパラメータ（続き）

パラメータ	説明
T_{ovlok}	**BCLK の立上がりエッジ後の BLOK 有効時間**
T_{ohlok}	**BCLK の立上がりエッジ後の BLOK ホールド時間**
T_{ovareq}	**BCLK の立上がりエッジ後の AREQ 有効時間**
T_{ohareq}	**BCLK の立上がりエッジ後の AREQ ホールド時間**

4.12 ASB デコーダ

AMBA システムのデコーダは、集中型アドレスデコーディング機能を実行するために使用します。これには主に 2 つの利点があります。

- 周辺ユニットをシステム・メモリマップから独立させることによって、それらの移植性を向上します。
- アドレスデコーディングおよびバス制御機能を一元化することによって、スレーブの設計を単純化します。

デコーダの 3 つの主要なタスクは以下の通りです。

- アドレスデコーダ
- デフォルトの転送応答
- プロテクション・ユニット

ASB デコーダは、ASB バス上の各スレーブの選択信号を生成し、ある状況下では、どのスレーブも選択せず、トランザクション応答自体を提供します。

デコーダはスレーブインターフェースを非常に単純化し、バス上で行なわれる可能性のある様々なタイプの転送についてスレーブが理解する必要性を無くします。

デコーダの重要な機能とは、アドレスデコーディングのために DECODE サイクルを提供することによって、システムの性能を向上することを可能にすることです。デコーダは、転送が SEQUENTIAL か、NONSEQUENTIAL かを認識することができるので、デコーダが、必要に応じて DECODE サイクルを追加だけ行うのは簡単なタスクとなります。

デコーダは、システムの性能を著しく向上させるのに実際に役立ちます。AMBA システム以外における危険なパスの例として、読み出し転送を以下に示します。

1. マスターからのアドレス出力
2. スレーブを選択するためのアドレスデコード
3. スレーブからバスマスターへのデータ出力および応答

しかし、AMBA システムでは、バスマスターが SEQUENTIAL 転送を実行しているときはいつでも中間段階を無くすことができます。これは、選択されているスレーブが前の転送と同じことが既知だからです。デコーダはこの事実を使用して、システムの性能を向上することができます。これは、必要に応じて、NONSEQUENTIAL 転送のためのアドレスデコーディングにウェイトステートを挿入するだけで可能になります。これは、DECODE サイクルの挿入として知られています。

アドレスデコーディングに追加のウェイトステートが必要とされない、クロック周波数が十分低い設計においては、デコーダの役割は単純になります。

デコーダはまた、多くのバスメンテナンス機能を提供するために使用されます。まず最初に、デコーダは、簡単なプロテクション・ユニットとして機能します。この装置はメモリマップの不正あるいは保護された領域にアクセスしようとするバスマスターに ERROR 応答を発行できます。デコーダはまた、スレーブが選択されていないとき、ADDRESS-ONLY 転送中に転送応答を返します。

4.12.1 インターフェースダイアグラム

図 4-37 には、ASB デコーダを示します。

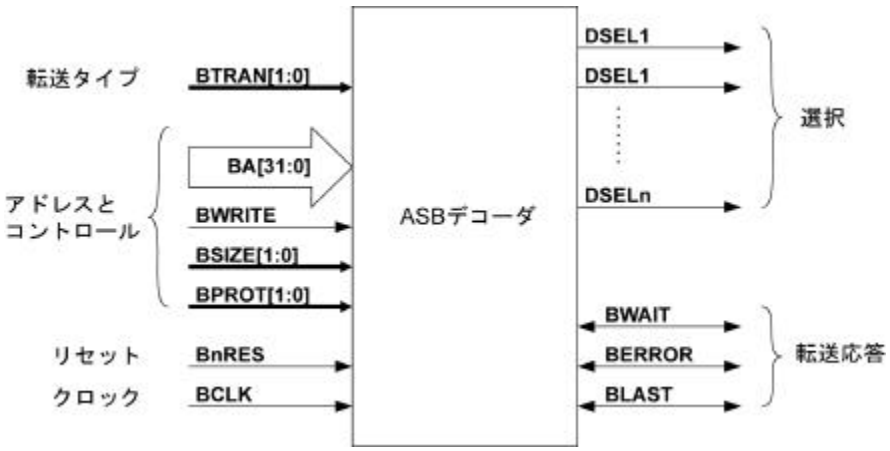


図 4-37 ASB デコーダのインターフェースダイアグラム

4.12.2 デコーダの説明

システム設計の性能要求に応じて、デコーダの実装には 2 つの方法が可能です。

- デコーダの通常の実装は、NONSEQUENTIAL 転送時の、およびメモリ境界を超えるバースト転送を分割するための、DECODE サイクルの挿入を組み込んでいます。
- 通常、クロック周波数が低い一部のシステム設計では、DECODE サイクルは必要とされず、従ってより簡単なデコーダが実装されます。

デコードサイクルがある場合

デコーダは、クロックの立下がりエッジから動作し、かつ 4 つの状態を持つステートマシンとして実装されます (図 4-38 参照)。リセット中は、ステートマシンは ADDRONLY 状態に入ります。

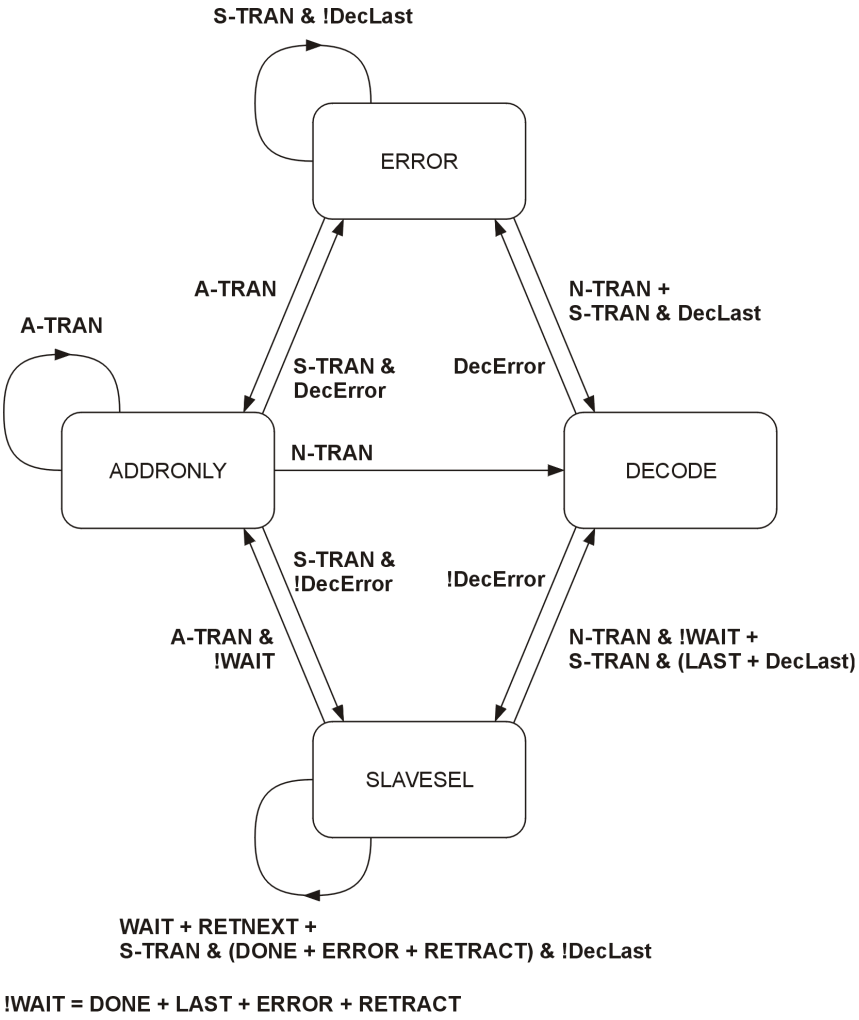


図 4-38 デコードサイクルがあるデコーダのステートマシン

ステートマシンにおける遷移は、次の転送の転送タイプ、現在の転送からの転送応答、および 2 つの内部デコーダ信号 **DecLast** と **DecError** によって制御されます。状態図に示す WAIT、DONE、LAST、ERROR、RETNEXT および RETRACT は、転送応答信号のエンコーディングに対応します。

DecLast は、SEQUENTIAL 転送がメモリ境界を越えようとしていることを検出したときに、デコーダが生成します。外部 **BLAST** 信号と組み合わせて使用して SEQUENTIAL 転送時でもアドレスのデコードを実行させます。

DecError は、別のデコーダ内部信号で、そしてデコーダが以下のことを検出したときに生成されます。

- 転送のアドレスにスレーブが存在しない。
- その転送が、保護されたメモリ領域に対してである。
- 転送の位置合わせがメモリシステムによってサポートされていない。

デコーダは、以下の機能を実行します。

- ADDRONLY 状態では、
  - アドレスを推測してデコードします。
  - **BCLK** の LOW フェーズ中、DONE 転送応答を返します。
  - 次の転送の転送タイプが S-TRAN で、そのアドレスが有効である場合、**BCLK** の HIGH フェーズ中に **DSELx** をアクティブにします。
- DECODE 状態では、
  - アドレスをデコードします。
  - **BCLK** の LOW フェーズ中、WAIT 転送応答を返します。
  - そのアドレスが有効である場合、**BCLK** の HIGH フェーズ中に **DSELx** をアクティブにします。
- SLAVESEL 状態では、
  - 転送応答が、選択されたスレーブによって駆動されます。
  - 転送が後回しになっている間、または次の転送が SEQUENTIAL で、かつ LAST 状態が検出されない場合、**DSELx** をアクティブ状態に保持します。
- ERROR 状態では、
  - **BCLK** の LOW フェーズ中、ERROR 転送応答を返します。

デコードサイクルがない場合

デコードサイクルを実装していないデコーダは、DECODE 状態が除去されます。これにより、図 4-39 に示すように、状態図が単純になります。

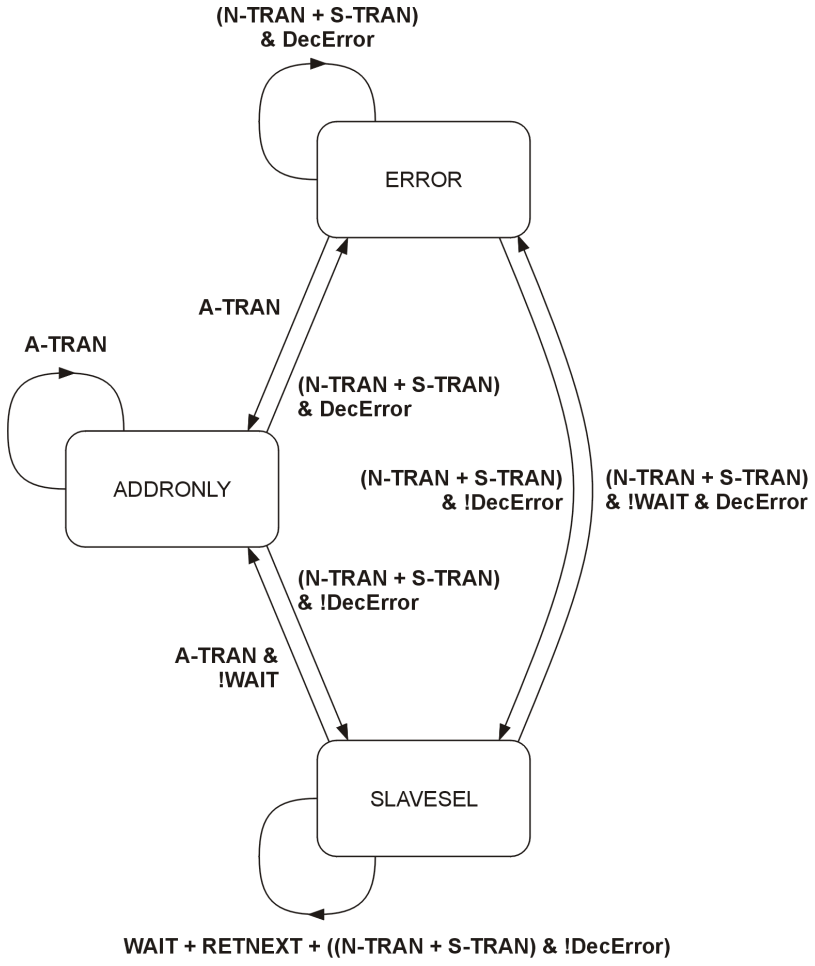


図 4-39 デコードサイクルのないデコーダのステートマシン

4.12.3 タイミングダイアグラム

DECODE サイクルがある ASB デコーダのタイミングパラメータを、図 4-40 に示します。DECODE サイクルがないデコーダのパラメータを、図 4-41 に示します。これら 2 つのダイアグラム間の主な相違点は、DECODE サイクルが挿入されないとき、**DSEL** 信号のタイミングがアドレスと制御信号のタイミングに依存するようになることです。

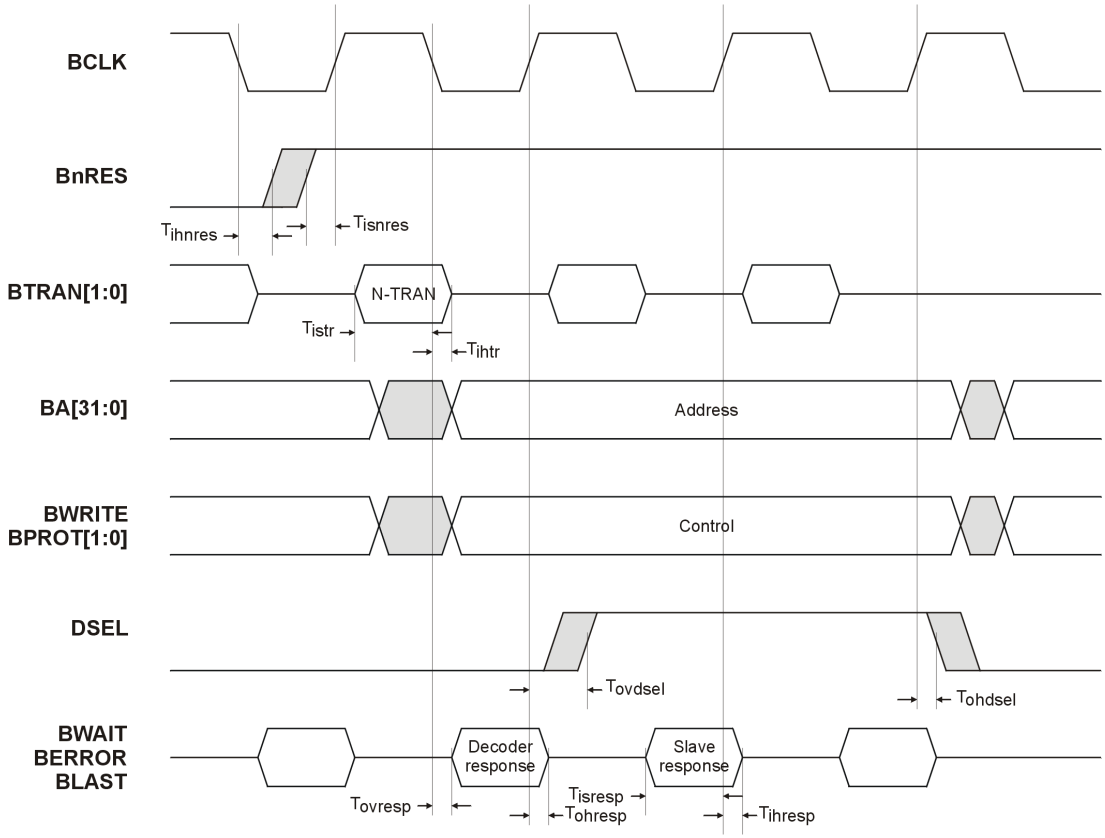


図 4-40 デコードサイクルのある ASB デコーダ

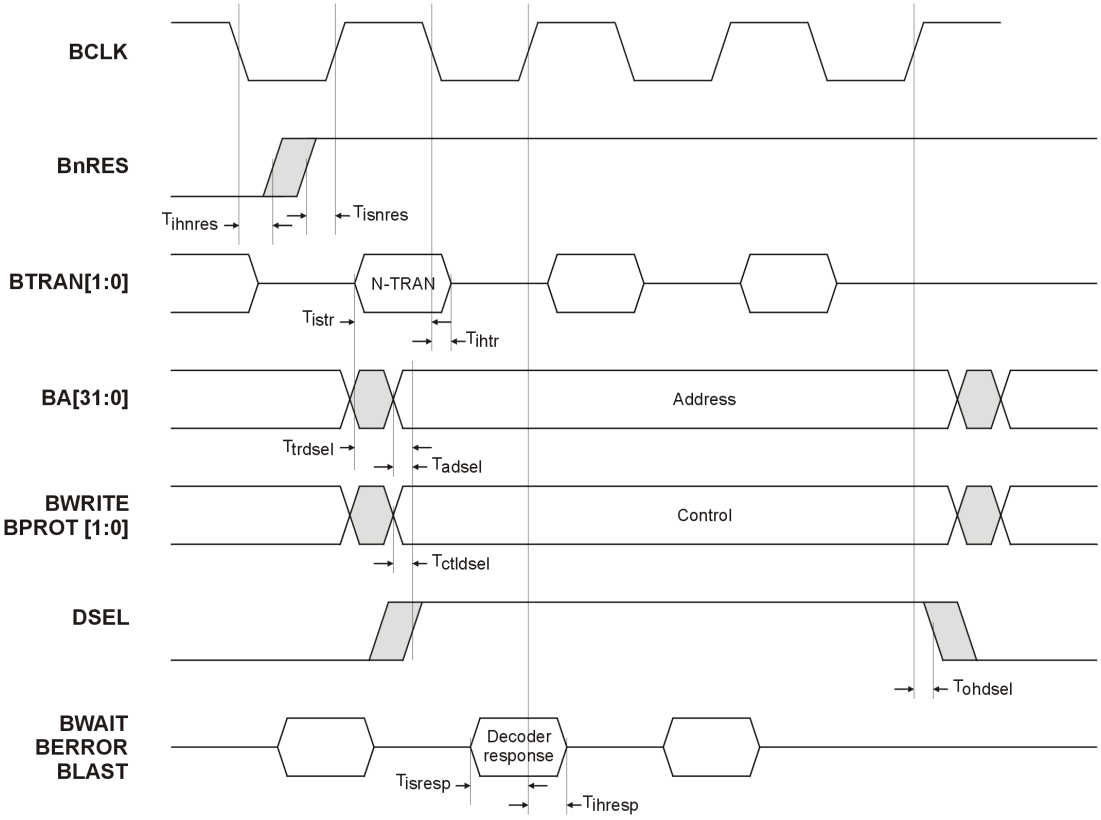


図 4-41 デコードサイクルのないASB デコーダ

4.12.4 タイミングパラメータ

ASB デコーダと関係があるタイミングパラメータを以下の表に示します。

- 入力信号については、表 4-11 に示します。
- 出力信号については、表 4-12 に示します。
- 組み合わせて生成された出力については、表 4-13 に示します。

表 4-11 ASB デコーダの入力パラメータ

パラメータ	説明
T_{clkl}	**BCLK** LOW **時間**
T_{clkh}	**BCLK** HIGH **時間**
T_{isnres}	**BCLK の立上がりエッジまでの BnRES インアクティブ設定時間**
T_{ihnres}	**BCLK の立下がりエッジ後の BnRES インアクティブホールド時間**
T_{istr}	**BCLK の立下がりエッジまでの BTRAN 設定時間**
T_{ihtr}	**BCLK の立下がりエッジ後の BTRAN ホールド時間**
T_{isresp}	**BCLK の立上がりエッジまでの BWAIT、BERROR、BLAST 設定時間**
T_{ihresp}	**BCLK の立上がりエッジ後の BWAIT、BERROR、BLAST ホールド時間**

表 4-12 ASB デコーダの出力パラメータ

パラメータ	説明
T_{ovresp}	**BCLK の立下がりエッジ後の BWAIT、BERROR、BLAST 有効時間**
T_{ohresp}	**BCLK の立上がりエッジ後の BWAIT、BERROR、BLAST ホールド時間**
T_{ovdsel}	**BCLK の立上がりエッジ後の DSEL 有効時間**
T_{ohdsel}	**BCLK の立上がりエッジ後の DSEL ホールド時間**

表 4-13 ASB デコーダの組合わせパラメータ

パラメータ	説明
T_{trdsel}	**BTRAN 有効から DSEL 有効までの遅延**
T_{adsel}	**BA 有効から DSEL 有効までの遅延**
$T_{ctldsel}$	**BWRITE 有効と BPROT[1:0] から DSEL 有効までの遅延**

4.13 ASB アービタ

AMBA システム内のアービタの役割は、どのマスターがバスにアクセスできるかを制御することです。すべてのバスマスターはアービタとの二本線の REQUEST および GRANT インターフェースを持ち、アービタはサイクルごとに、優先順位付け機構を使用して、バスを要求しているバスマスターのうち、優先順位が現在最も高いのはどのマスターかを判断します。

共用バスロック信号 **BLOK** は、現在認可されているバスマスターによって駆動され、現在の転送が次の転送と分割不可で、かつ他のいかなるマスターにもバスの使用を認可すべきでないことを知らせるために使用します。

優先順位方式の詳細は規定されていません。それはアプリケーションごとに定義されます。アービタが AMBA もしくは AMBA 以外での他の信号を使用して、使用中の優先順位方式に影響を及ぼすことは容認されます。

4.13.1 インターフェースダイアグラム

図 4-42 には、ASB アービタの信号インターフェースを示します。

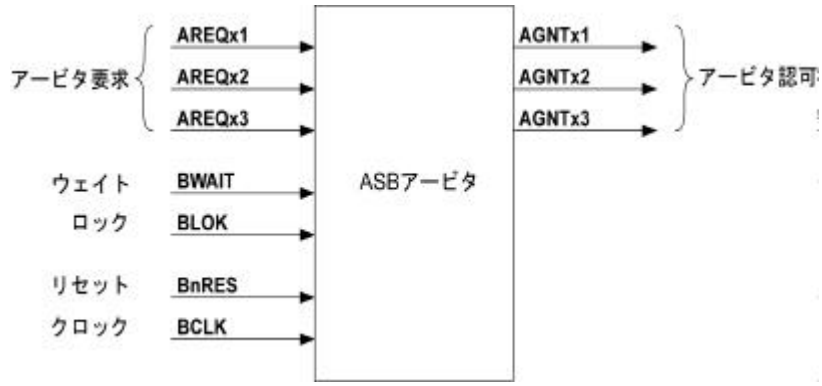


図 4-42 ASB アービタのインターフェースダイアグラム

4.13.2 アービタの説明

バスはクロックサイクルごとにアービトレーションを受けます。アービタは **BCLK** の立下がりエッジで、すべての要求信号 **AREQx** を取得し、**BCLK** の LOW フェーズ中に、内部優先順位方式および **BLOK** の値を使用して適切な **AGNTx** 信号を有効にします。

アービトレーションはサイクルごとに変化可能なため、延長された転送中で、その転送の最終完了以前に最優先順位のバスマスターが何度か切り換わることはあり得ます。転送が完了したときにアクティブになる **AGNT** を持っているバスマスターは、次にアクティブなバスマスターになります。

バスマスター・ハンドオーバー中は、**BLOK** 信号は駆動されず、したがってアービタはこの信号が LOW であることを仮定する必要があります。

アービタは、現在認可されているマスターのコピーを保有する必要があり、以下のことが可能です。

- **BLOK** がアクティブになると、現在のバスマスターを認可状態に保持します。
- バスマスターがいつ切り換わるか判断し、バスマスター・ハンドオーバーサイクルがいつ行なわれるかを判断します。

4.13.3 タイミングダイアグラム

図 4-43 には、アービタのタイミングパラメータを示します。

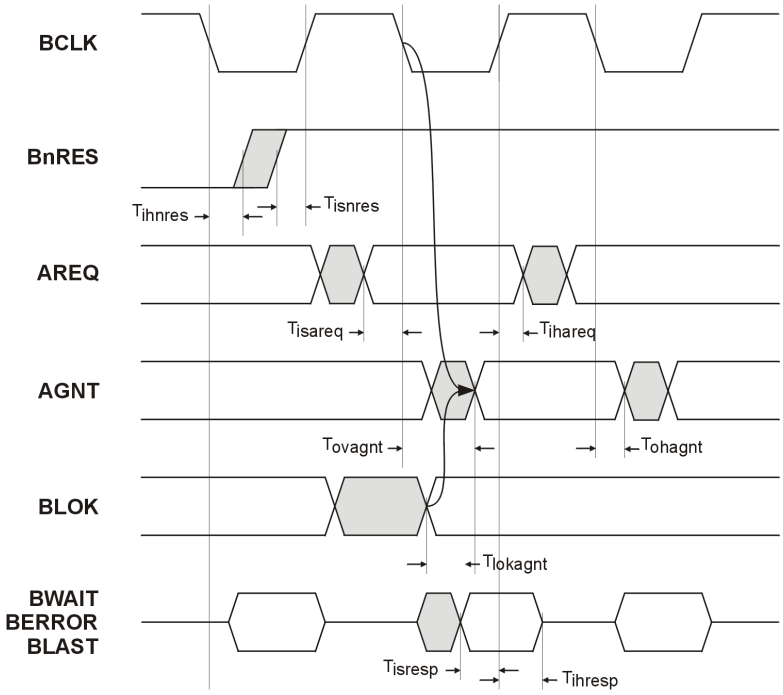


図 4-43 ASB アービタのパラメータ

4.13.4 タイミングパラメータ

ASB アービタと関係があるタイミングパラメータを以下の表に示します。

- 入力信号については、表 4-14 に示します。
- 出力信号については、表 4-15 に示します。
- 組み合わせて生成された出力については、表 4-16 に示します。

表 4-14 ASB アービタの入力パラメータ

パラメータ	説明
T_{clkl}	**BCLK** LOW **時間**
T_{clkh}	**BCLK** HIGH **時間**
T_{isnres}	**BCLK の立上がりエッジまでの BnRES インアクティブ設定時間**
T_{ihnres}	**BCLK の立下がりエッジ後の BnRES インアクティブホールド時間**
T_{isareq}	**BCLK の立下がりエッジまでの AREQ 設定時間**
T_{ihareq}	**BCLK の立上がりエッジ後の AREQ ホールド時間**
T_{isresp}	**BCLK の立上がりエッジまでの BWAIT 設定時間**
T_{ihresp}	**BCLK の立上がりエッジ後の BWAIT ホールド時間**

表 4-15 ASB アービタの出力パラメータ

パラメータ	説明
T_{ovagnt}	**BCLK の立下がりエッジ後の AGNT 有効時間**
T_{ohagnt}	**BCLK の立下がりエッジ後の AGNT ホールド時間**

表 4-16 ASB アービタの組合わせパラメータ

パラメータ	説明
$T_{lokagnt}$	**BLOK 有効から AGNT 有効までの遅延**

Chapter 5
AMBA APB

本章では、以下のセクションで、*アドバンスド・マイクロコントローラバス・アーキテクチャ* (AMBA) の *アドバンスド・ペリフェラルバス* (APB) の仕様について説明します。

5.1 AMBA APB について

アドバンスド・ペリフェラルバス (APB) は、*アドバンスド・マイクロコントローラバス・アーキテクチャ* (AMBA) バスの階層の一部で、消費電力を最小にし、インターフェースの複雑さを低減するように最適化されています。

AMBA APB は、周波数帯域が小さく、パイプライン方式バスインターフェースの高性能を必要としない周辺ユニットとのインターフェースをとるのに使用すべきです。

最新バージョンの APB は、すべての信号遷移がクロックの立上がりエッジによってのみ決まることを保証します。この改良により、APB 周辺ユニットをどんな設計フローにも容易に統合でき、以下の利点が得られます。

- 高周波数動作時の性能が改善されました。
- 性能がクロックのマーク／スペース比に依存していません。
- シンプル・クロック・エッジの使用によりスタティックなタイミング解析が簡単になりました。
- 自動テスト挿入について特別に考慮する必要がありません。
- 多くの*特定用途向け IC* (ASIC) ライブラリには、より優れた立上がりエッジレジスタが選択されています。
- サイクルベースシミュレータとの統合が容易です。

APB に対するこれらの変更により、APB と新しいアドバンスド・ハイパフォーマンスバス (AHB) 間の接続もより容易になります。

5.1.1 代表的な AMBA ベースマイクロコントローラ

AMBA ベース・マイクロコントローラは通常、外部メモリ周波数帯域を維持できる高性能システム*中枢*バスから構成されています。このバスには CPU とその他の*ダイレクト・メモリアクセス* (DMA) 装置が接続されています。さらに、より低い周波数帯域の周辺ユニットが配置されているより狭い APB バスへのブリッジも接続されています。図 5-1 に、代表的な AMBA システムの APB を示します。

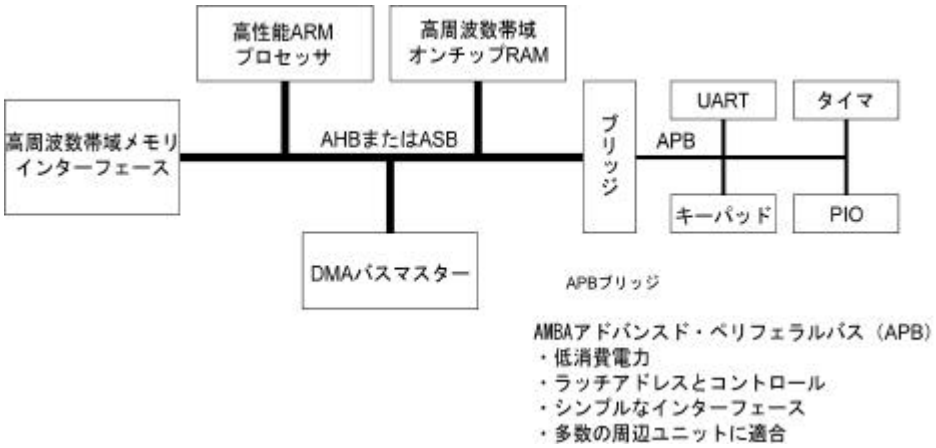


図 5-1 代表的な AMBA システムにおける APB

テストインターフェース・コントローラ (TIC) を組み込んだシステムバスにより、システムバスと APB モジュール両方のモジュール試験が可能になります。

5.2 APB 仕様書

APB 仕様書は、以下の見出しで説明しています。

- 状態図
- 書込み転送：P.5-5
- 読出し転送：P.5-6

5.2.1 状態図

図 5-2 に状態図を示しますが、これを使用して周辺バスの動作を表わすことができます。

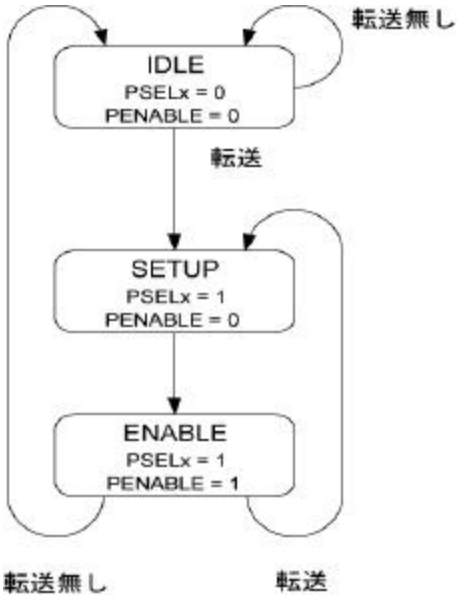


図 5-2 状態図

ステートマシンの動作は、以下に説明する 3 つの状態を介しています。

IDLE　周辺バスのデフォルト状態。

SETUP　転送が必要になったとき、バスは SETUP 状態に入り、適切な選択信号 **PSELx** がアクティブになります。バスは、1 クロックサイクルの間 SETUP 状態のままで、クロックの次の立上がりエッジ時には必ず ENABLE 状態になります。

ENABLE　　ENABLE 状態では、イネーブル信号 **PENABLE** がアクティブになります。アドレス、書込み、および選択信号はすべて、SETUP 状態から ENABLE 状態への遷移中には安定しています。
ENABLE 状態もまた、1 クロックサイクルの間だけ継続し、この状態後は、それ以上の転送が必要なければバスは IDLE 状態に戻ります。あるいはまた、別の転送が続くことになっている場合、バスが直接、SETUP 状態に入ります。
アドレス、書込み、および選択信号が、ENABLE 状態から SETUP 状態への遷移中のグリッチ発生を容認できます。

5.2.2 書込み転送

図 5-3 には、基本的な書込み転送を示します。

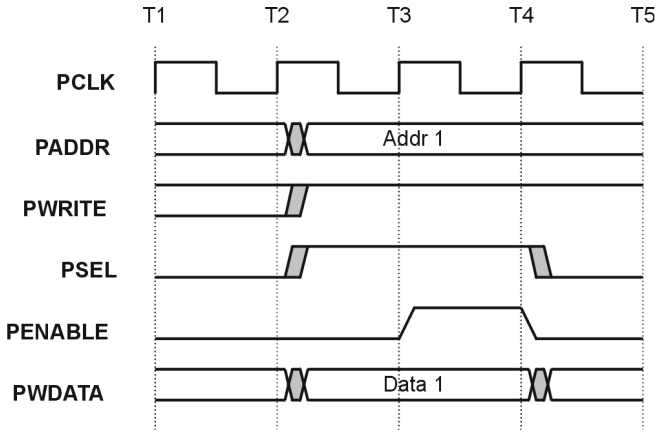


図 5-3 書込み転送

書込み転送は、アドレス、書込みデータ、書込み信号、および選択信号から始まり、これらはすべてクロックの立上がりエッジ後に変化します。転送の最初のクロックサイクルは、SETUP サイクルと呼ばれます。次のクロックエッジ後に、イネーブル信号 **PENABLE** がアクティブになります。これは ENABLE サイクルが行なわれていることを示します。アドレス、データ、および制御信号はすべて、ENABLE サイクル中ずっと有効のままです。転送はこのサイクルの終了時に完了します。

イネーブル信号 **PENABLE** は転送の終了時にインアクティブになります。その転送の直後に同じ周辺ユニットへの別の転送が続かない限り、選択信号も LOW になります。

電力消費低減のため、アドレス信号と書込み信号は、転送後に次のアクセスが行なわれるまで変化しません。

プロトコルはイネーブル信号についてのみ完全な遷移を必要とします。連続した転送の場合、選択信号と書込み信号がグリッチを起こすことがあります。

5.2.3 読出し転送

図 5-4 には、読出し転送を示します。

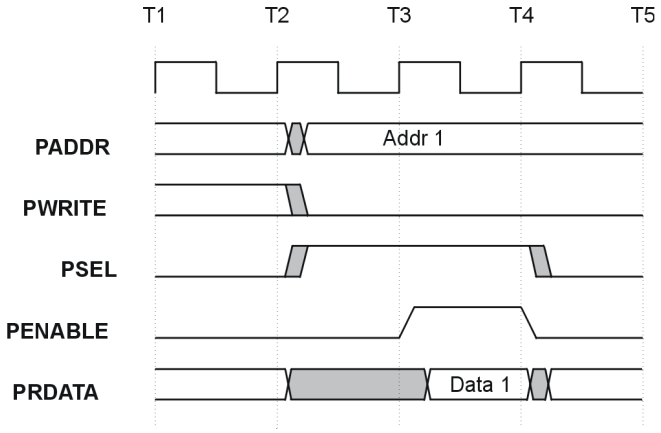


図 5-4 読出し転送

アドレス、書込み、選択、およびストローブ信号のタイミングは、書込み転送の場合とまったく同じです。読出しの場合、ENABLE サイクル中、スレーブはデータを供給しなければなりません。そのデータは、ENABLE サイクル終了時のクロックの立上がりエッジで取得します。

5.3 APB AMBA コンポーネントについて

タイミングパラメータに対して以下の表記法を使用します。

- T_{is} - 入力設定時間
- T_{ih} - 入力ホールド時間
- T_{ov} - 出力有効時間
- T_{oh} - 出力ホールド時間

5.4 APB ブリッジ

APB ブリッジは AMBA APB 上の唯一のバスマスターです。その上、APB ブリッジはより高いレベルのシステムバス上ではスレーブでもあります。

5.4.1 インターフェースダイアグラム

図 5-5 には、APB ブリッジの APB 信号インターフェースを示します。

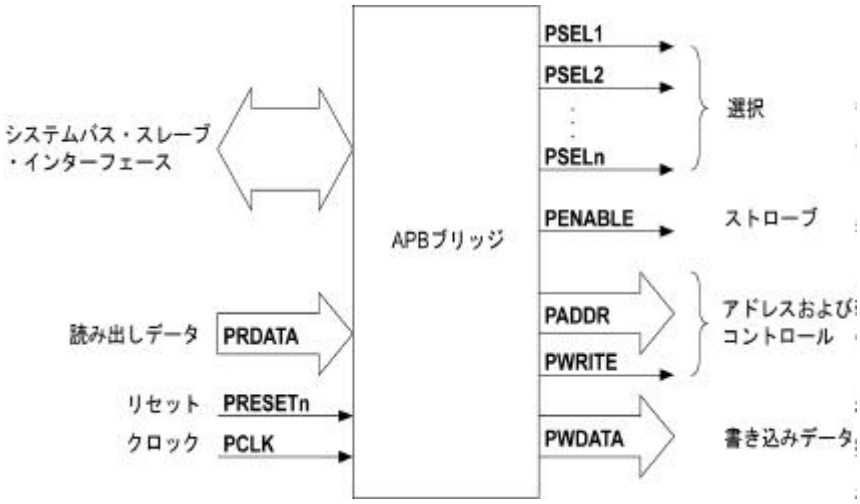


図 5-5 APB ブリッジのインターフェースダイアグラム

5.4.2 APB ブリッジの説明

ブリッジ装置はシステムバス転送を APB 転送に変換し、以下の機能を果たします。

- アドレスをラッチし、転送の間中、有効を維持します。
- アドレスをデコードし、周辺ユニット選択信号 **PSELx** を生成します。転送中、選択信号は 1 つだけアクティブになります。
- 書込み転送の場合、データを APB に送ります。
- 読出し転送の場合、APB データをシステムバスに送ります。
- 転送のタイミングストローブ信号 **PENABLE** を生成します。

5.4.3 タイミングダイアグラム

図 5-6 には、APB ブリッジのタイミングパラメータを示します。

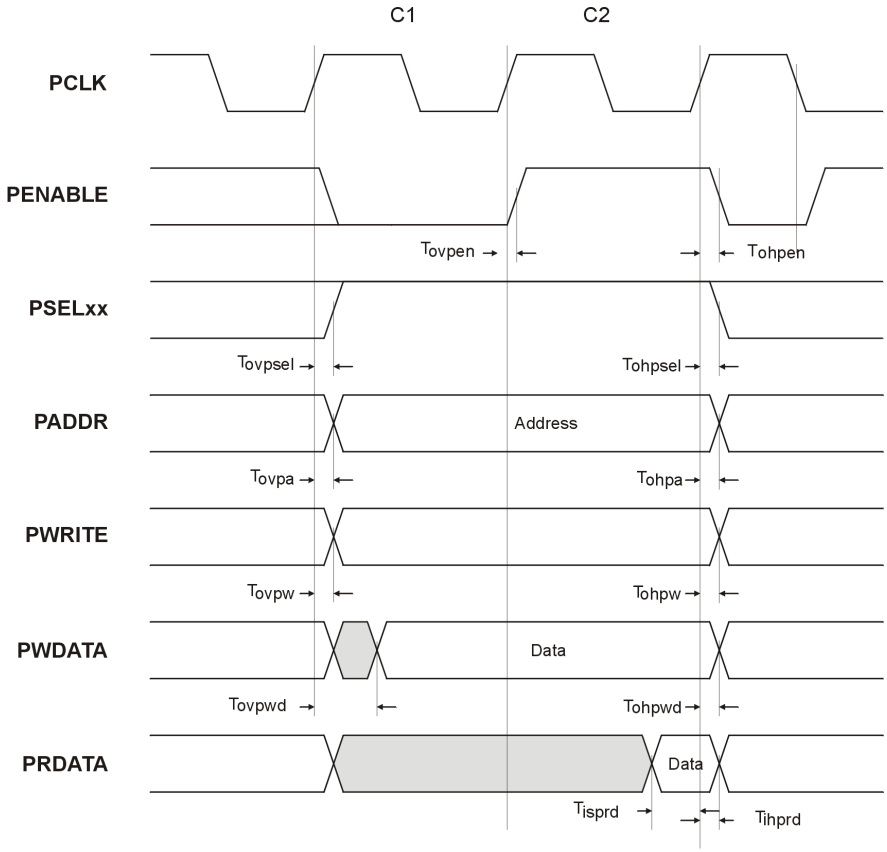


図 5-6 APB ブリッジの転送

5.4.4 タイミングパラメータ

APB ブリッジ関連のタイミングパラメータを、入力信号について表 5-1 に示し、出力信号について表 5-2 に示します。

表 5-1 APB ブリッジの入力パラメータ

パラメータ	説明
T_{clkl}	**PCLK** LOW **時間**
T_{clkh}	**PCLK** HIGH **時間**
T_{isnres}	**PCLK の立上がりエッジまでの PRESETn インアクティブ設定時間**
T_{ihnres}	**PCLK の立上がりエッジ後の PRESETn インアクティブホールド時間**
T_{isprd}	**読出し転送の場合、PCLK の立上がりエッジまでの PRDATA 設定時間**
T_{ihprd}	**読出し転送の場合、PCLK の立上がりエッジ後の PRDATA ホールド時間**

表 5-2 APB ブリッジの出力パラメータ

パラメータ	説明
T_{ovpen}	**PCLK の立上がりエッジ後の PENABLE 有効時間**
T_{ohpen}	**PCLK の立上がりエッジ後の PENABLE ホールド時間**
T_{ovpsel}	**PCLK の立上がりエッジ後の PSEL 有効時間**
T_{ohpsel}	**PCLK の立上がりエッジ後の PSEL ホールド時間**
T_{ovpa}	**PCLK の立上がりエッジ後の PADDR 有効時間**
T_{ohpa}	**PCLK の立上がりエッジ後の PADDR ホールド時間**
T_{ovpw}	**PCLK の立上がりエッジ後の PWRITE 有効時間**
T_{ohpw}	**PCLK の立上がりエッジ後の PWRITE ホールド時間**
T_{ovpwd}	**書込み転送の場合、PCLK の立上がりエッジ後の PWDATA 有効時間**
T_{ohpwd}	**書込み転送の場合、PCLK の立上がりエッジ後の PWDATA ホールド時間**

5.5 APB スレーブ

APB スレーブはシンプルですが、柔軟なインターフェースです。このインターフェースの的確な実装は採用されている設計様式に依存し、多くの様々なオプションが可能です。

5.5.1 インターフェースダイアグラム

図 5-7 には、APB スレーブの信号インターフェースを示します。

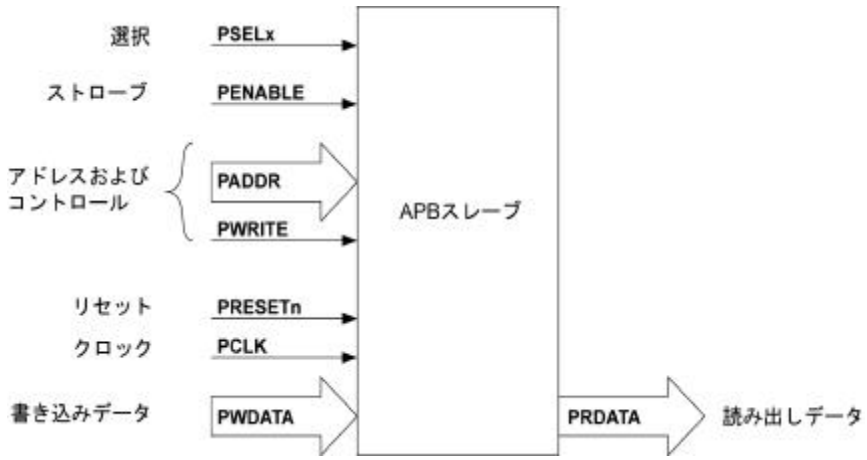


図 5-7 APB スレーブインターフェースの説明

5.5.2 APB スレーブの説明

APB スレーブインターフェースは非常に柔軟です。

書込み転送の場合、そのデータを以下の時点でラッチします。

- **PSEL** が HIGH の時の、**PCLK** の立上がりエッジ
- **PSEL** が HIGH の時の、**PENABLE** の立上がりエッジ

選択信号 **PSELx**、アドレス **PADDR**、および書込み信号 **PWRITE** を組み合わせて、書込み動作によって更新すべきレジスタを判断することができます。

読出し転送の場合、**PWRITE** が LOW で、**PSELx** と **PENABLE** の両者が HIGH のとき、データをデータバス上に送ることができます。一方、**PADDR** は、どのレジスタを読み出すべきかを判断するのに使われます。

5.5.3 タイミングダイアグラム

APB バススレーブへのアクセス関連のタイミングパラメータを図 5-8 に示します。

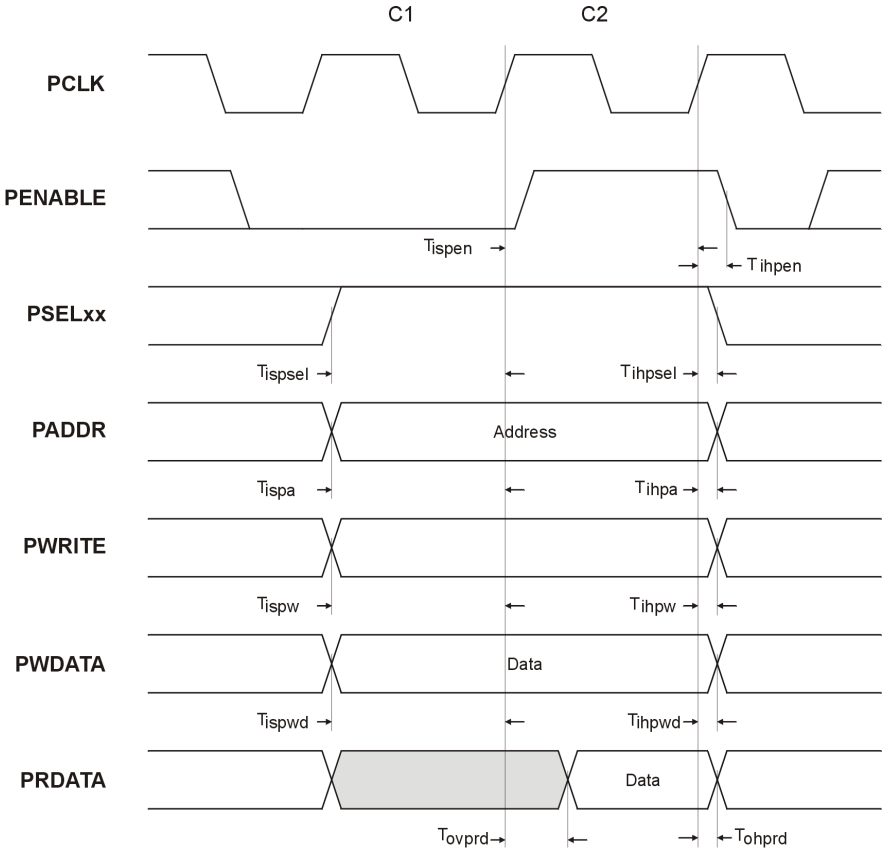


図 5-8 APB スレーブ転送

5.5.4 タイミングパラメータ

APB スレーブ関連のタイミングパラメータで、入力信号を表 5-3 に示し、出力信号を表 5-4 に示します。

表 5-3 APB スレーブの入力パラメータ

パラメータ	説明
T_{clkl}	**PCLK** LOW **時間**
T_{clkh}	**PCLK** HIGH **時間**
T_{isnres}	**PCLK の立上がりエッジまでの PRESETn インアクティブ設定時間**
T_{ihnres}	**PCLK の立下がりエッジ後の PRESETn インアクティブホールド時間**
T_{ispen}	**PCLK の立上がりエッジまでの PENABLE 設定時間**
T_{ihpen}	**PCLK の立上がりエッジ後の PENABLE ホールド時間**
T_{ispsel}	**PCLK の立上がりエッジまでの PSEL 設定時間**
T_{ihpsel}	**PCLK の立上がりエッジ後の PSEL ホールド時間**
T_{ispa}	**PCLK の立上がりエッジまでの PADDR 設定時間**
T_{ihpa}	**PCLK の立上がりエッジ後の PADDR ホールド時間**
T_{ispw}	**PCLK の立上がりエッジまでの PWRITE 設定時間**
T_{ihpw}	**PCLK の立上がりエッジ後の PWRITE ホールド時間**
T_{ispwd}	**書込み転送の場合、PCLK の立上がりエッジまでの PWDATA 設定時間**
T_{ihpwd}	**書込み転送の場合、PCLK の立上がりエッジ後の PWDATA ホールド時間**

表 5-4 APB スレーブの出力パラメータ

パラメータ	説明
T_{ovprd}	**読出し転送の場合、PCLK の立上がりエッジ後の PRDATA 有効時間**
T_{ohprd}	**読出し転送の場合、PCLK の立上がりエッジ後の PRDATA ホールド時間**

5.6 APB と AHB との接続

AMBA APB と AHB との接続を以下のセクションで説明します。

- *読出し転送*
- *書込み転送*：P.5-16
- *連続転送*：P.5-18
- *トライステート・データバスの実装：P.5-19*

5.6.1 読出し転送

図 5-9 では、読出し転送を説明します。

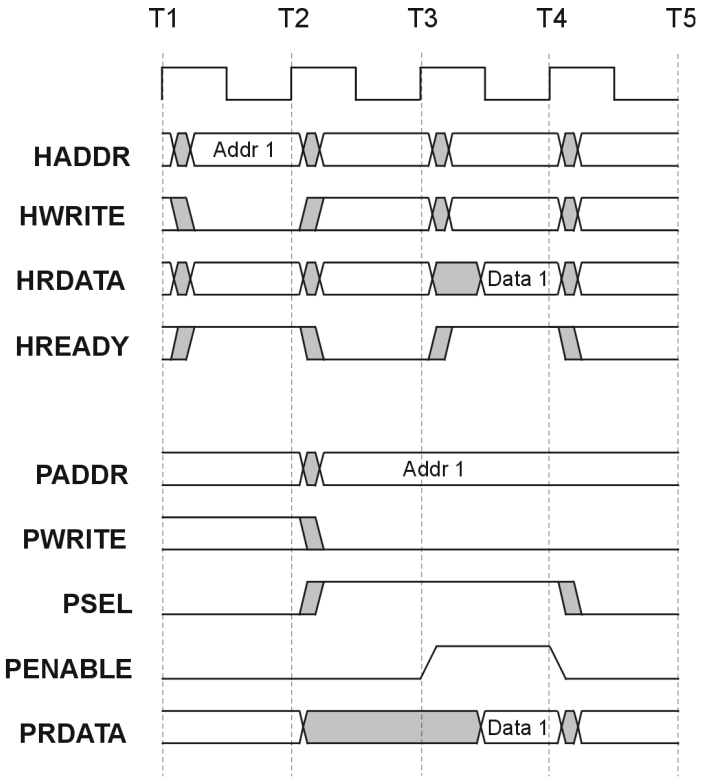


図 5-9 AHB への読出し転送

転送は AHB 上で時間 T1 において開始し、そのアドレスは APB によって T2 において取得されます。その転送が周辺バスに対するものであれば、このアドレスが送信され、適切な周辺ユニット選択信号が生成されます。周辺バス上のこの最初のサイクルは、SETUP サイクルと呼ばれます。これは **PENABLE** 信号がアクティブになると、ENABLE サイクルを伴います。

ENABLE サイクル中、周辺ユニットは読出しデータを提供しなければなりません。通常、この読出しデータを直接 AHB に返すことが可能です。AHB 上では、バスマスターが ENABLE サイクルの終了時のクロックの立上がりエッジでデータを取得します。これは図 5-9 の時間 T4 において行なわれます。

クロック周波数が非常に高いシステムでは、ブリッジが ENABLE サイクルの終了時に読出しデータを記録し、その後に続くサイクルで AHB バスマスターにこれを返す必要があります。これは、周辺バスの読出し転送の場合、追加のウェイトステートを必要としますが、AHB がより高いクロック周波数での実行が出来、その結果、システム性能の総合的な向上をもたらします。読出し転送のバーストを図 5-10 に示します。読出し転送はすべて、1 つのウェイトステートを必要とします。

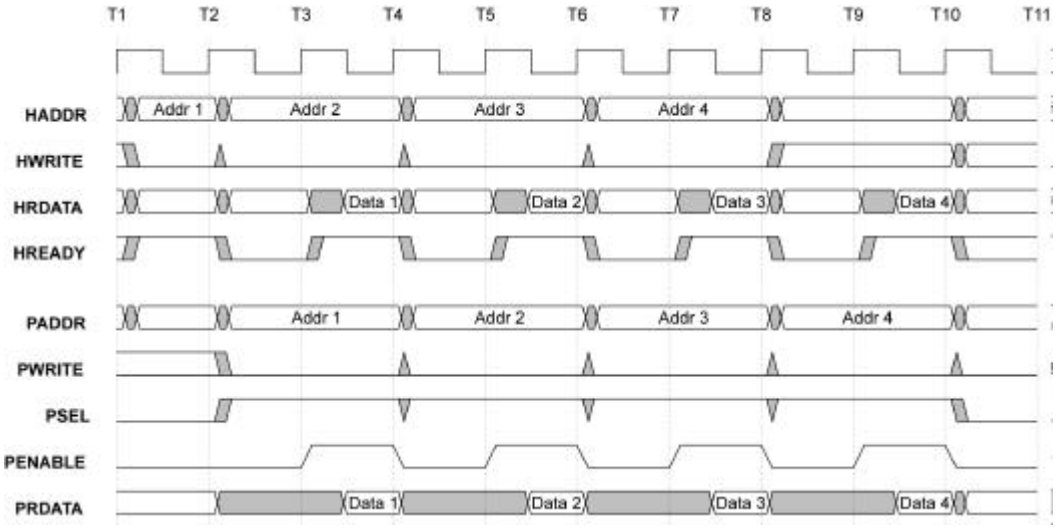


図 5-10 読出し転送のバースト

5.6.2 書込み転送

図 5-11 には、書込み転送を示します。

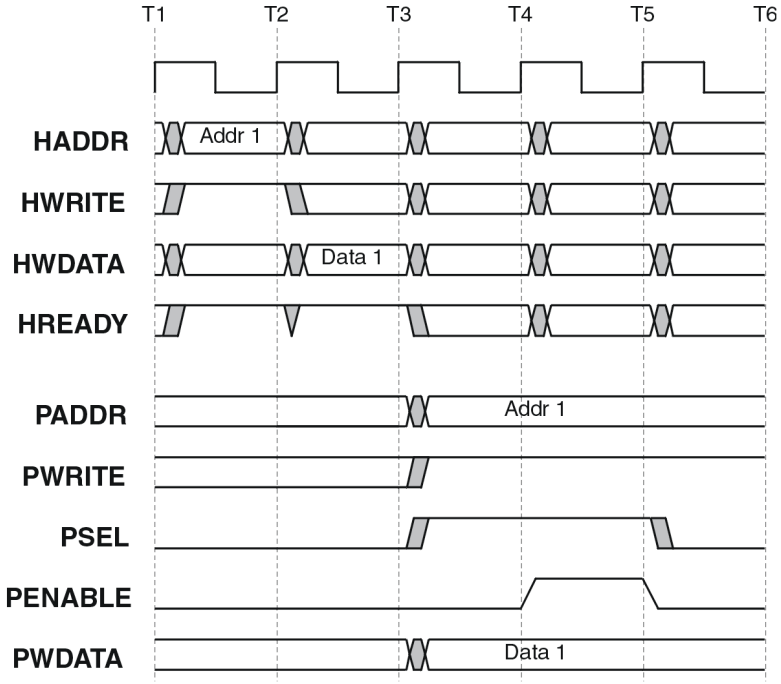


図 5-11 AHB からの書込み転送

APB への単一書込み転送は、ウェイトステート無しで行なうことができます。ブリッジは、転送のアドレスとデータを取得し、その折 APB 上の書込み転送の間、これらの値を保持します。

書込み転送のバーストを図 5-12 に示します。

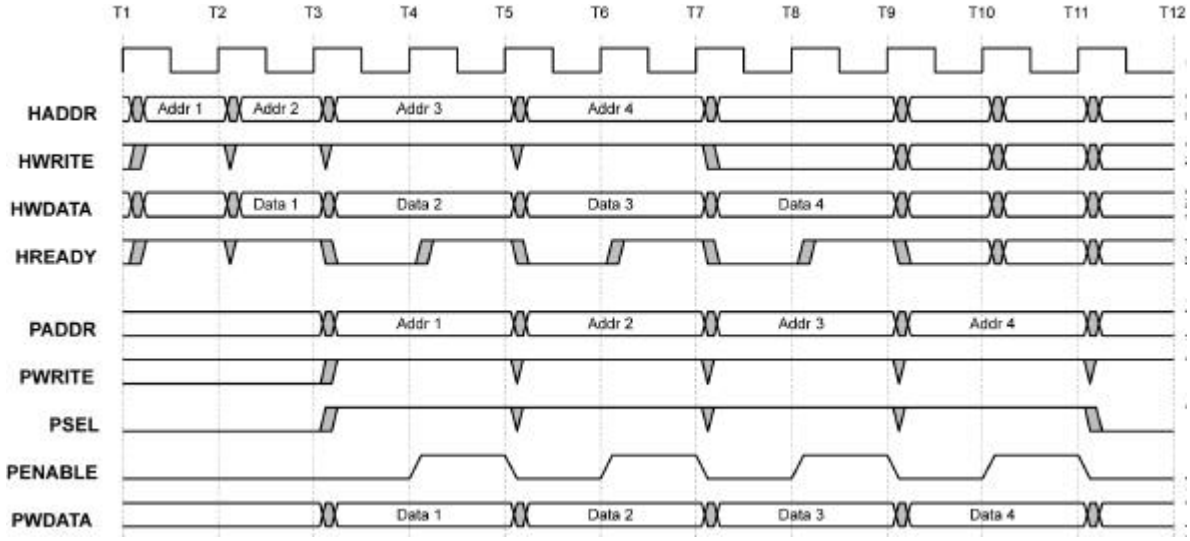


図 5-12 書込み転送のバースト

最初の転送はウェイトステート無しで完了できますが、その後の周辺バスへの転送は、各転送ごとに 1 つのウェイトステートを必要とします。

現在の転送が周辺バス上で続いているときに、ブリッジが次の転送のアドレスを取得するためには、ブリッジがアドレスレジスターを 2 つ組み込むことが必要です。

5.6.3 連続転送

図 5-13 に、多くの連続転送を示します。シーケンスは書込みで始まり、その後に読出し、書込み、読出しと続きます。

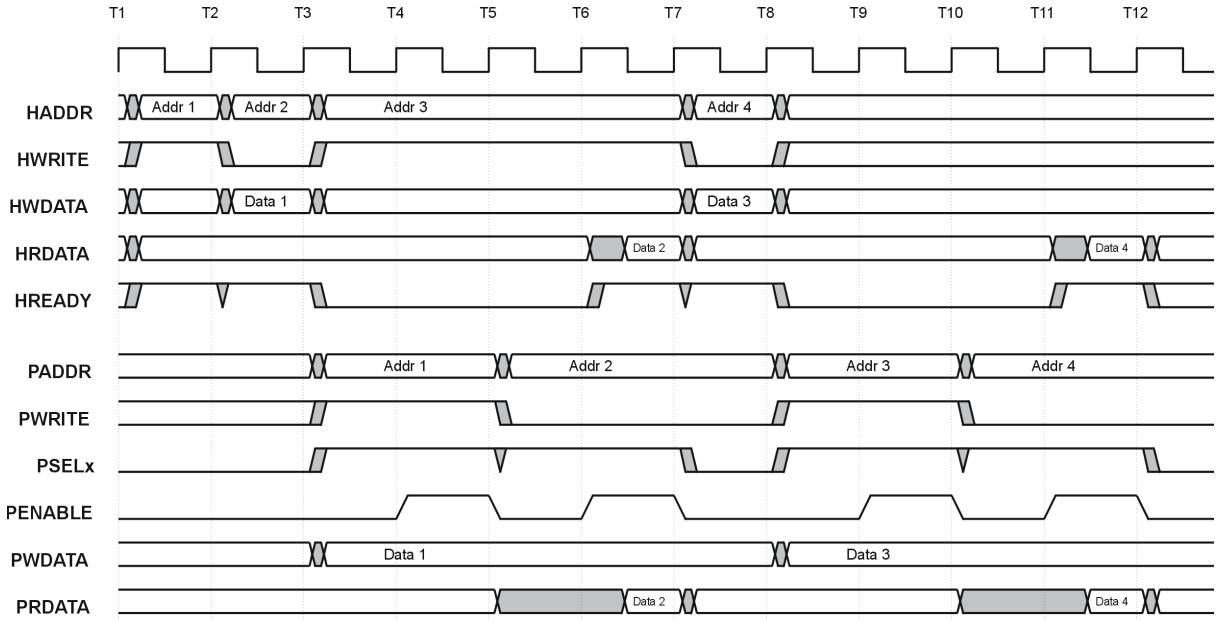


図 5-13 連続転送

読出し転送が書込み転送の直後に続いている場合、その読出しを完了するために 3 つのウェイトステートが必要であることが図 5-13 から分かります。実際、プロセッサベースの設計では、読出しを伴う書込みは頻繁には行なわれません。その理由は、プロセッサがその 2 つの転送間で命令取出しを行ない、命令メモリが APB 上に存在することは考えられないからです。

5.6.4 トライステート・データバスの実装

AMBA APB を、独立した読出しおよび書込みデータバスと共に実装することを推奨します。これにより多重バス方式または OR バス方式を使用して、APB 上の各種スレーブを相互接続することが可能になります。トライステートバスを使用した場合、読出しデータと書込みデータが同時には発生しないので、読出しおよび書込みデータバスを 1 つのバスにまとめることができます。

図 5-14 では、トライステートバスを使用してデータバスを実装する場合、特別な配慮が必要がないこと説明します。データバスが読出し転送の SETUP サイクルでトライステートである場合、およびバスが IDLE 状態にあるときはいつでも、ターンアラウンドの 1 クロックサイクル分が異なったデータドライバ間に発生します。書込み転送バーストの場合、ブリッジが各転送の SETUP サイクルでデータを駆動するので、ターンアラウンドは行なわれません。しかし、これは完全に容認できます。その理由は、ブリッジが書込み転送用データバスの唯一のドライバであり、そのためにターンアラウンド時間が必要ないからです。

図 5-14 では、読出しおよび書込みデータバスを 1 つのトライステート・データバスにうまくまとめる方法を示します。

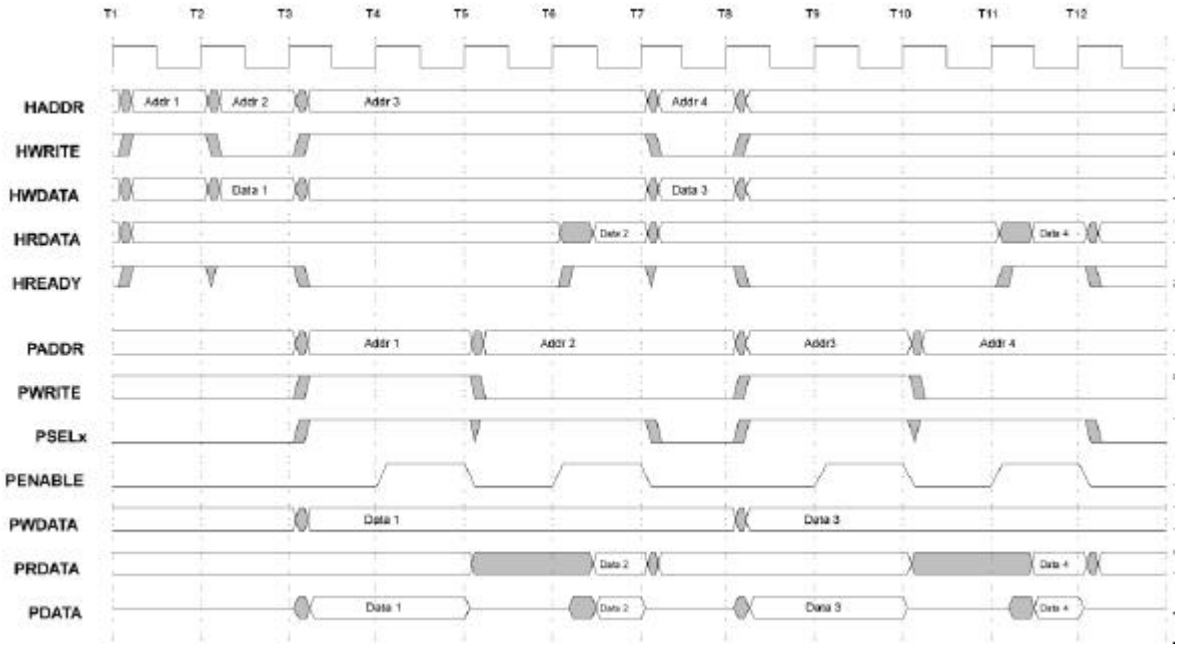


図 5-14 トライステート・データバス

5.7 APB と ASB との接続

AMBA APB と ASB との接続を以下のセクションで説明します。

- *書込み転送*
- *読出し転送：P.5-21*

5.7.1 書込み転送

図 5-15 では、ASB から APB へのインターフェースを構築する方法を説明します。書込みのバーストには追加のウェイトステートが必要であるが、書込み転送はウェイトステート無しで行なうことができます。

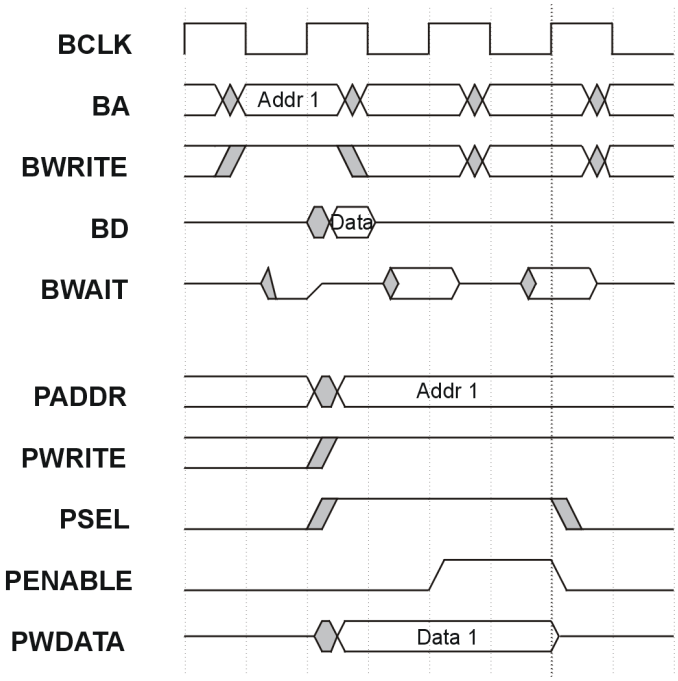


図 5-15 ASB からの書込み転送

5.7.2 読出し転送

読出し転送は常に 1 つのウェイトステートを必要とします (図 5-16 参照)。クロック周波数が高いシステムでは、読出しデータがブリッジを通過するのに十分な時間を持ち、かつ APB 上で有効になることを保証するために、追加のウェイトステートを挿入することが必要です。

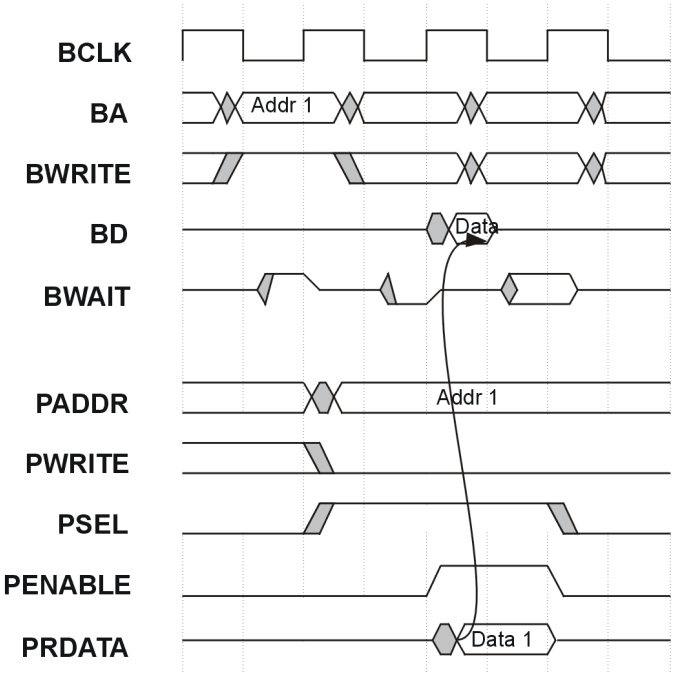


図 5-16 ASB への読出し転送

5.8 第 D 版 APB 周辺ユニットと第 2.0 版 APB との接続

第 2.0 版仕様書に合わせて設計された一部の周辺ユニットと、それより前の改訂版に合わせて設計された他の周辺ユニットを組み合わせて使用する場合、第 2.0 版ブリッジを使用し、それより前のバージョンの周辺ユニットはその新ブリッジと共に使用するように改造することを推奨します。

本セクションでは、第 D 版の 1 つの周辺ユニットを最新版の APB に合わせて改造する方法を示します。改造しなければならない周辺ユニットがたくさんある場合は、1 つの集中ブロック内で改造を行なうことがより効率的です。

第 D 版と第 2.0 版 APB 仕様間の根本的相違は以下の 2 つです。

- イネーブル信号に対するストローブ信号のタイミング
- 読出しデータを取得するポイント

ある周辺ユニットが第 D 版仕様に合わせて設計されているか、それとも第 2.0 版仕様に基づいて設計されているかをすばやく判断するためには、その装置が **PSTB** 入力を持っている (この場合、第 D 版仕様) か、それとも **PENABLE** 入力を持っている (この場合、第 2.0 版仕様) かを確認して下さい。図 5-17 には、既存の第 D 版周辺ユニットとのインターフェースをとるために必要な 2 つの段階を示します。

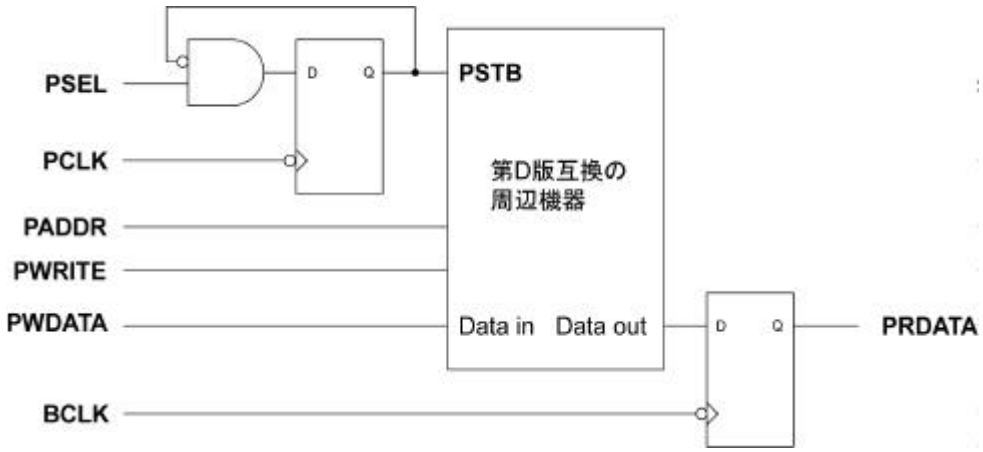


図 5-17 第 D 版周辺ユニットの接続

まず初めに、**PSEL** 信号を使用して、**PSTB** 信号を生成します。**PSTB** の帰還信号を使用して、その信号が 1 クロックサイクルの間アクティブになることを保証することができます。

場合により必要になる 2 番目のインターフェース段階は、周辺ユニットからの出力データ (読出しデータ) に関する立下がりエッジトリガー式レジスタまたは透過ラッチです。これは、周辺ユニットが立下がりエッジ後に出力データを変更する場合にのみ必要です。

Chapter 6
AMBA 試験方法

本章では、AMBA モジュール設計と共に使用されるテストインターフェースについて説明します。本書は、以下のセクションから構成されています。

6.1 AMBA テストインターフェースについて

AMBA 試験方針により、システム内の個別のモジュールを単独でテストすることができます。各モジュールは、バスからの転送だけを使ってテストできるように設計されており、他のシステム要素の介在は当てにしていません。したがって、バスに直接接続されていない、周辺ユニットの入力と出力にアクセスできることが必要です。これはテストハーネスによって提供されます。

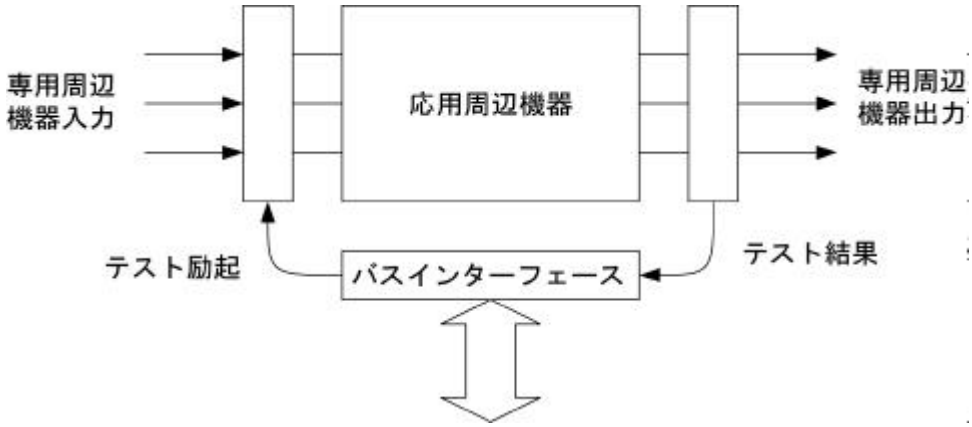


図 6-1 周辺ユニットのテストハーネス

ゲート数が少ない*テストインターフェース・コントローラ* (TIC) のバスマスターモジュールがシステムに必要です。これにより、外部から加えられたテストベクトルを内部バス転送に変換することができます。

TIC はできるだけ小さい 3 線式ハンドシェイクメカニズムを使用して、テストベクトルの適用を制御します。*外部バスインターフェース* (EBI) のデータ経路は高速 32 ビット・パラレル・ベクトルインターフェースを提供するために使用されます。

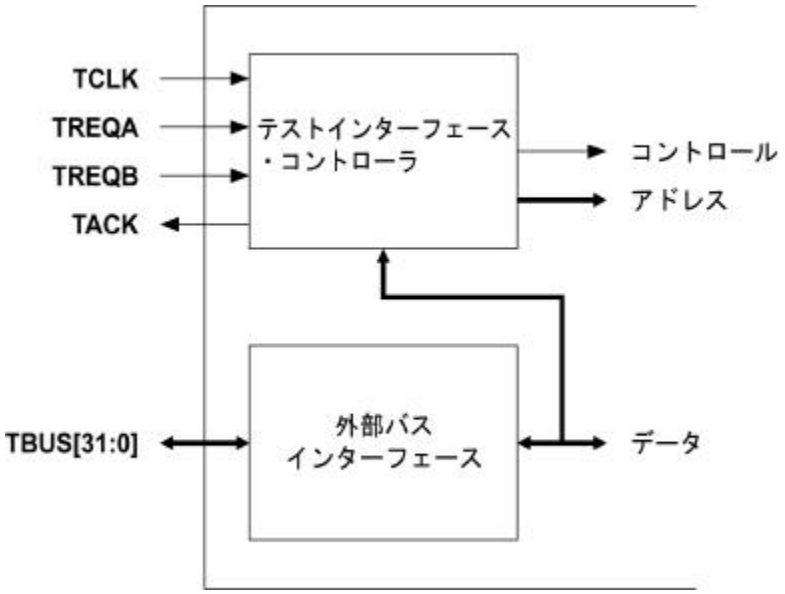


図 6-2 TIC と外部バスインターフェースの相互作用

このテストベクトル適用方法をサポートするためには、32 ビット双方向ポートがテストアクセス中に使用できなければなりません。32 ビットの外部データバス・インターフェースを搭載したシステムの場合は、これは簡単明瞭です。16 ビットおよび 8 ビット・データバス設計は、例えば、16 または 24 アドレスラインがテストモード・アクセス用の双方向テストポート信号として再構成されることを必要とします。

6.2 外部インターフェース

外部テストインターフェースは、以下の要素から構成されます。

- テストクロック
- 3 つの制御信号
- 32 ビット・テストバス

テストモードの開始および終了を制御するためには、2 つの専用信号ピン (**TREQA** および **TACK**) だけが必要です。残りの信号は、既存のデバイスピンを再使用することによって提供することができます。

6.2.1 テストバス要求 A

TREQA はテストバス要求 A 入力信号で、専用デバイスピンとして要求されます。

通常のシステム動作中、**TREQA** 信号はテストモードの開始を要求するために使用されます。これによりテストバスがトライステート状態になり、テストベクトルを適用することができます。

テスト中、本信号は、**TREQB** と組み合わせて使用され、次のサイクルで適用されるテストベクトルの種類を示します。

6.2.2 テストバス要求 B

TREQB はテストバス要求 B 入力信号です。

テスト中、本信号は、**TREQA** と組み合わせて使用され、次のサイクルで適用されるテストベクトルの種類を示します。

6.2.3 テスト応答

TACK は、テストバス応答出力信号で、専用デバイスピンとして要求されます。

テストバス応答信号は、テストバスの使用を認可されたことを外部に通知し、いつテストアクセスが完了したかも知らせます。**TACK** が LOW のとき、現在のテストベクトルは、**TACK** が HIGH になるまで延長されなければなりません。**TREQA** と **TREQB** 信号は、**TACK** が HIGH のときだけ、TIC によって取得されます。

表 6-1 および表 6-2 には、**TREQA**、**TREQB**、および **TACK** 信号の動作を示します。これらの信号は、テストモードが開始したか否かによってその機能が異なります。

表 6-1 通常動作中のテスト制御信号

TREQA	TREQB	TACK	説明
0	0	0	通常動作
1	0	0	テストモード開始要求
0	1	0	予備 (外部マスター要求用)
-	-	1	テストモード中

表 6-2 テストモード中のテスト制御信号

TREQA	TREQB	TACK	説明
-	-	0	現アクセス未完了
1	1	1	アドレスベクトル、制御ベクトル、あるいはターンアラウンド・ベクトル
1	0	1	書込みベクトル
0	1	1	読出しベクトル
0	0	1	テストモード終了

6.2.4 テストクロック

TCLK はテストクロック入力信号です。

テストモードでは、内部バスクロックは外部 **TCLK** ソースから駆動されます。このピンは通常のクロック発振器ソース入力またはポート置換信号です。システムバス・クロックはノーマルモードとテストモードの切換え時に、グリッチを起こしてはいけません。

テストモードの開始時に、TIC は **TACK** 信号をアクティブにすることにより、テストクロック入力に切り換えられたことを知らせます。

6.2.5 テストバス

TBUS[31:0] は 32 ビット双方向テストポートです。

テストバスは、アドレス、制御、および書込みベクトルを適用するための入力として使用されます。テストインターフェース・プロトコルは、テストバスの方向を変える際には必ずターンアラウンド時間が与えられることを保証します。

6.3 テストベクトルの種類

テストインターフェース関連のテストベクトルには 5 種類あります。

- アドレスベクトル
- 書込みベクトル
- 読出しベクトル
- 制御ベクトル
- ターンアラウンド・ベクトル .

アドレスベクトル、制御ベクトル、およびターンアラウンド・ベクトルはすべて、**TREQA** と **TREQB** 信号において同じ値で示されます。以下のルールを使用して、どの種類のベクトルが適用されているかを判断できます。

- 単一のアドレス / 制御ベクトルが適用されているとき、それはアドレスベクトルです。
- アドレス / 制御ベクトルのバーストが適用されているとき、制御ベクトルである最後のベクトルは別として、それらはすべてアドレスベクトルです。
- 読出しベクトル、または読出しベクトルのバーストはいつもターンアラウンド・ベクトルを伴います。これは、ターンアラウンド・ベクトルが一つだけ出現するときです。テストインターフェースの ASB バージョンはターンアラウンド・ベクトルをただ 1 つ必要としますが、AHB バージョンは 2 つ必要とします。

6.4 テストインターフェース・コントローラ

テストインターフェース・コントローラ (TIC) は、外部テストバス **TBUS[31:0]** からテストベクトルを受け入れ、バス転送を開始するバスマスターです。TIC はアドレスベクトルをラッチし、必要に応じてアドレスをインクリメントして、テストベクトルの読出しおよび書込みバーストを許可します。

6.4.1 テスト転送パラメータ

テストモード開始時のデフォルトの TIC バスマスター動作は、以下の通りです。

- 32 ビット転送幅
- 特権システムアクセス

これは多くの組込み型システム設計をテストするには十分で、オンチップ・テストサポート・ロジックを最小化します。上記制御信号のダイナミックな変化を必要とするシステムの場合、制御ベクトルメカニズムが使用され、TIC 内の制御信号を更新します。

制御ベクトルのビット 0 は、その制御ベクトルが有効であるかどうかを示すために使用されます。したがって、ビット 0 が LOW の状態で制御ベクトルが適用された場合、そのベクトルは無視され、制御情報を更新しません。このメカニズムにより、多くのサイクルの間、制御情報を更新することなく、ビット 0 が LOW であるアドレスベクトルが適用できます。

6.4.2 インクリメンタルアドレス指定

テストインターフェースを使用して、バーストアクセスをサポートするために、TIC はバスアドレスのインクリメントをサポートします。インクリメントされるアドレスビット数は、テストインターフェースを経由して要求される最大バーストアクセス長に左右されます。これはシステムに依存しますが、代表的な実装では 8 ビットアドレスインクリメンタが使用され、ワード転送を使用して 1kB 境界までのバーストアクセスが可能です。

制御ベクトルは、TIC 内でアドレスインクリメンタを有効および無効にするメカニズムも提供します。これにより、内部 RAM のテストに使用されるインクリメンタルアドレスへのバーストアクセスが可能になります。一方、アドレスインクリメントは無効にすることができます。その結果、バーストの連続アクセスが同じアドレスに対して行なわれます。これは 1 つの周辺レジスタから頻繁に読み出すために必要です。

転送サイズが動的に変化する場合、バーストモード・アクセスに対するアドレスインクリメンタ・サポートはバイト、ハーフワード、およびワードオフセット単位のインクリメントをサポートできなければなりません。そのため、適応型アドレスインクリメンタ・ロジックが必要です。

アドレスインクリメンタは、デフォルトでは無効で、使用前に制御ベクトルを用いて有効にしなければなりません。

6.4.3 テストモードの開始

通常の動作モードでは、**TREQA** は LOW になり、テストアクセスが要求されていないことを示します。通常、外部バスインターフェースの一部となるテストバスが、必要に応じて通常動作に使用されます。テストモードを開始することにより、内部バスでの転送を発生させるテストベクトルを外部から適用できます。

テストモードを開始するために、以下のシーケンスが必要です。

1. テストバス・アクセスを要求するために、**TREQA** をアクティブにします。
2. TIC が内部バスを認可されたときテストモードが開始します。これは、**TACK** 信号のアクティブ化によって示されます。
3. このポイントでは、**TCLK** が内部クロック信号のソースになります。
4. テストモードが開始すると、アドレスベクトルを起動するために、**TREQB** がアクティブになります。

TIC は、有効なアドレスベクトルが適用されるまで、内部転送を行ないません。

同期テスターは、バスの **TACK** をポーリングするようになっていません。通常、**TREQA** は、バスへのアクセス権を確実に取得出来るように、最小限のサイクル数間アクティブになります (最長のウェイトステートにある周辺ユニット・アクセスの終了、または現行命令を終了させるための全バス・マスターへの最大サイクル数のバスにおいて)。

6.4.4 アドレスベクトル

アドレスベクトルは、読出しあるいは書込み動作が行なわれる前に適用されなければなりません。アドレスベクトルを適用するために、以下のシーケンスが必要です。

1. **TREQA** と **TREQB**が両方とも HIGH になり、アドレスベクトルの次サイクルを示します。
2. 次のサイクルでは、アドレスが **TBUS[31:0]** に適用されます。一方、**TREQA** および **TREQB** は、次のテストベクトルの種類を反映するために変化します。

一部の高速システムでは、アドレスが外部テストバスから内部アドレスバスまで伝播するための十分な時間を確保するために、2 つ以上のアドレスベクトルを連続して適用することが必要です。このような場合、TIC はアドレスベクトルの最初のサイクルの間 **TACK** を否定し、アドレスベクトルの別のサイクルが適用されるようにします。

6.4.5 制御ベクトル

制御ベクトルは常に、アドレスベクトルのシーケンスにおける最後のベクトルで、TIC 内の制御情報を更新するために使用されます。このシーケンスは、以下の通りです。

1. **TREQA** と **TREQB** は HIGH になり、アドレスベクトルの次のサイクルを示します。
2. 次のサイクルでは、アドレスが **TBUS[31:0]** に適用されます。次のサイクルで制御ベクトルが発生するので、**TREQA** と **TREQB** は両方とも HIGH のままです。
3. 次のサイクルでは、制御情報が **TBUS[31:0]** に適用されます。一方、**TREQA** および **TREQB** は、次のテストベクトルの種類を反映するために変化します。
4. 最終的に、内部バス上で転送が行なわれます。

制御ベクトルのビット 0 を LOW に設定することによって、無効な制御ベクトルを適用することができます。これは TIC 内の制御情報を変化させません。

6.4.6 書込みテストベクトル

テストモードがいったん首尾よく開始すると、読出しおよび書込み動作がテストインターフェースを通して実行できます。書込み動作を内部的に実行するためには、アドレス、続いて書込みデータを供給することが必要です。

書込み転送に使用されるアドレスは、直前のベクトルに依存し、書込みベクトルは、以下のうちのいずれかの後に発生します。

- 単一のアドレスベクトル
- アドレス / 制御ベクトルのシーケンス
- 書込みのバーストを形成する、別の書込みテストベクトル
- 単一の読出し又は読出しのバーストの後のターンアラウンド・ベクトル

ウェイトステートの挿入によって内部バス転送が延長されると、これは外部的には **TACK** 信号が LOW になって示されます。後回し状態の間、**TREQA** と **TREQB** は、現在のベクトルが完了したときの後に続くベクトルタイプを示すのに変化します。しかし、書込みベクトルの場合、データが **TBUS[31:0]** に適用され続けることに注意することが重要です。

6.4.7 読出しテストベクトル

書込みテストベクトルと同様に、読出しテストベクトルは、以下に列挙する数多くの様々なベクトルの後に続きます。転送に使用されるアドレスは、直前のベクトルによって異なります。

- 単一のアドレスベクトル
- アドレス / 制御ベクトルのシーケンス
- 読出しのバーストを形成する、別の読出しテストベクトル
- 単一の書込みあるいは書込みのバースト

読出し、あるいは読出しのバーストは常に、外部 **TBUS** 信号に関するバスの衝突を防止するため、ターンアラウンド・ベクトルを伴います。書込みベクトルについて言えば、内部転送が延長されると、これは外部的には **TACK** 信号が LOW になって示されます。読出しデータは、内部転送が完了するまで、外部から取得してはいけません。

6.4.8 バーストベクトル

複数の書込みベクトルまたは読出しベクトルを結合して、ベクトルのバーストを形成することができます。これは、各読出しまたは書込みベクトルにアドレスベクトルを関連付ける必要を無くすことにより、テストベクトルがより高速に適用できるようにします。

バースト転送は、有効になっているアドレスインクリメンタを TIC が組み込んでいるか否かによって、インクリメント式アドレスまたはスタティックアドレスにのどちらかを使用します。アドレスインクリメンタが無い場合、TIC は一定アドレスへのノンシーケンシャル転送を実行します。

TIC が有効なアドレスインクリメンタを組み込んでいる場合は、各連続転送に使用されるアドレスは、転送サイズによって指示される数だけインクリメントされます。

6.4.9 バースト方向の変更

バーストの転送方向を、読出しから書込みへ、あるいは書込みから読出しへ変更することができます。

読出しから書込みへ変更する場合は、まだターンアラウンド・ベクトルを挿入することが必要です。これは新しいアドレスをロードしないが、内部的には新しいバーストを開始させます。これにより、内部スレーブは、バーストの方向が変わったことを知ることができます。

6.4.10 テストモードの終了

テストモードは、以下のシーケンスを使用して終了します。

1. 1 サイクルのアドレスベクトルを適用します。これは、内部転送が完了したことを保証します。
2. **TREQA** と **TREQB** の両方が LOW になり、テストモードが終了することを示します。
3. テストインターフェースが通常のシステム動作に設定されたとき、**TACK** は LOW になり、テストモードが終了したことを示します。

システム動作中に診断テストが実行できるようにテストモードが手際よく開始かつ終了できることが重要です。

6.5 AHB テストインターフェース・コントローラ

以下の状態図で、TIC の動作を説明します。

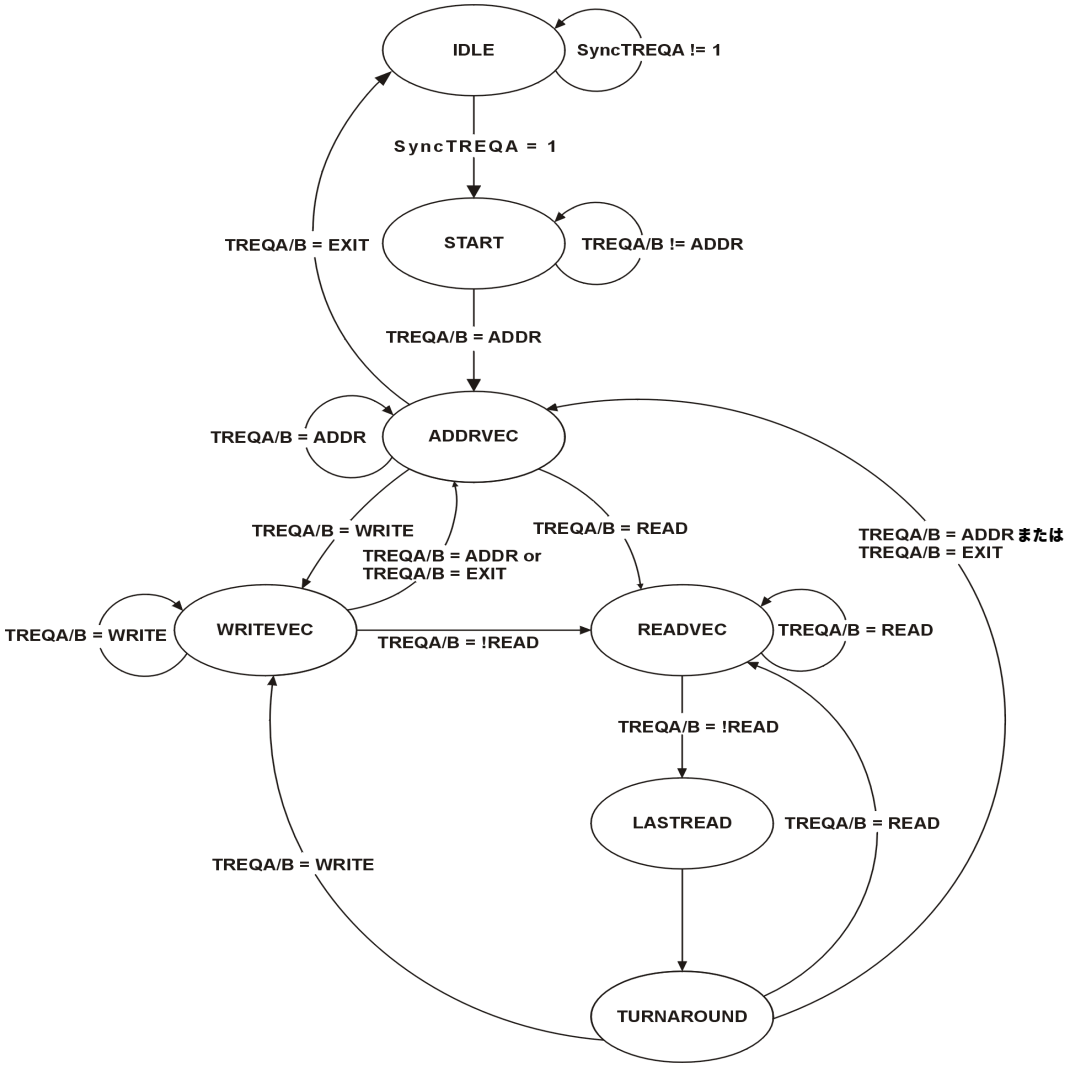


図 6-3 テストインターフェース・コントローラの状態図

TIC 状態図の動作を、以下に説明します。

- リセット時、TIC は IDLE 状態で、AHB の使用を要求しません。IDLE 状態にあるときは、**TACK** は LOW になり、テストインターフェースが使用できないことを示します。
- IDLE から START への遷移を除く、ステートマシンの全トランザクションを制御するために **TACK** 信号が使用されます。その他のあらゆる場合、**TACK** 信号が LOW であれば、ステートマシンは同じ状態のままです。
- **TREQA** 信号は、IDLE 状態から START 状態に移行するために使用されます。これは以前の仕様から変更されました。以前の仕様では、**TREQA** が HIGH で、**TREQB** が LOW であることが求められました。この新しい仕様には、通常の動作からテストモードへ移行するために、**TREQA** を使用できるという利点があります。
- 一部のシステム実装では、外部クロックソースから、テストモード中に使用される外部クロック **TCLK** へ切り換えることが必要です。**TREQA** が最初に HIGH になったとき、これはクロックソースを変更すべきであるという指示として使用できます。クロックの切換えがいつ首尾よく行なわれたかを示すリターン信号は、テストクロックが使用されるまで START 状態への移行を防止するために使用できます。
- クロック切換えが使用されている場合、テストモードが開始するまで **TREQA** がオンチップクロックに非同期であることがあり得ます。したがって、IDLE 状態から START 状態への移行を制御するための **TREQA** の同期バージョンを生成するために同期回路が使用されます。
- START 状態は、最初のベクトルがアドレス初期化前の読出しおよび書込みベクトルの発生を防止するアドレスベクトルであることを保証するために使用されます。START 状態は、**TREQA/B** がアドレスベクトルを示し、次の状態が ADDRVEC であるときにのみ終了します。
- ADDRVEC 状態では、TIC は **TBUS** にアドレスを記録します。ADDRVEC 状態はアドレスと制御ベクトルの両方に対して使用され、したがって、**TBUS** 上の値をアドレスベクトルとみなすべきか、あるいは制御ベクトルとみなすべきかを判断するための、追加のロジックが必要です。前のサイクルがアドレスベクトルであったが、後に続くサイクル (**TREQA/B** によって示される) がアドレスベクトルでない場合、現在のサイクルは制御ベクトルとなります。
- 数多くのサイクルの間、ADDRVEC 状態にとどまることは可能ですが、アドレスベクトルは通常、読出しか書込み転送のいずれかを伴います。
- 書込み転送が実行されている場合、TIC は、バス上の転送を開始すると同時に、WRITEVEC 状態に移行します。複数の書込み転送は、WRITEVEC 状態にとどまることによって実行することができます。通常、WRITEVEC はアドレスベクトルを伴いますが、READVEC 状態に移行することによって、読出し転送に直接、移行することもできます。

- 読出し、あるいは読出しのバーストが実行されるとき、TIC は READVEC 状態に入ります。この状態は、TIC がバス上の読出し転送を開始していることを示し、次のサイクルになって初めて読出しデータが現われます。READVEC 状態に最初に入ったとき、**TBUS** はトライステート状態になるが、READVEC 状態ではそれ以上のサイクルの間は駆動状態になります。
- すべての読出しベクトルは、2 つのターンアラウンド・ベクトルを伴わなければなりません。これらのうち最初のサイクルの場合、TIC は LASTREAD 状態に移行します。その間に、転送の最後の読出しが完了し、外部 **TBUS** 上に送られます。LASTREAD 状態の間、どの内部転送も開始せず、TIC はバス上で IDLE 転送を実行します。
- LASTREAD 状態に続いて、TIC は TURNAROUND 状態に移行します。その間、外部 **TBUS** はトライステート状態です。TURNAROUND 状態は通常、アドレスベクトルを伴うが、直ちに書込みベクトルまたは別の読出しに移行することもできます。
- テストを終了する通常の方法は、ADDRVEC 状態に戻り、その後 IDLE に戻り完全にテストを終了するために、**TREQA/TREQB** の両方を LOW に設定することです。実際、どのポイントにおいても **TREQA** と **TREQB** の両方を LOW に設定してテストモードを終了できます。これにより TIC がテストを終了します。

Note

TIC ベクトルを適用した場合は、**HLOCK** 出力をアクティブにし、その後テストを終了することは理論的には可能です。これが発生し、その後通常動作のもとで TIC がバスの使用を認可されると、そのバスが完全にロックされます。この発生を防止する保護手段は、TIC 内部に搭載されていません。

6.5.1 制御ベクトル

実行できる転送のタイプを判断するために、制御ベクトルが TIC 内部に組み込まれています。この制御ベクトルは **HSIZE**、**HPROT**、および **HLOCK** の値を設定するために使用されます。

テストモード開始時のデフォルトの TIC バスマスター動作は、以下の通りです。

- 32 ビット転送幅 - **HSIZE[1:0]** はワード転送を示します。
- 特権システムアクセス - **HPROT[3:0]** はキャッシュ可能な、およびバッファ可能な特権データアクセスを示します。

制御ベクトルのビット 0 は、制御ベクトルが有効かどうかを示すために使用されます。したがって、制御ベクトルがビット 0 を LOW として印加された場合、そのベクトルは無視され、制御情報を更新しません。このメカニズムにより、ビット 0 が LOW になるアドレスベクトルは多くのサイクルの間、制御情報を更新しないで適用されます。

デフォルトの設定は多くの組込み型システム設計をテストするには十分ですが、制御ベクトルは、転送の制御信号を変更するためや、および TIC が固定アドレスを生成すべきか、またはインクリメント式アドレスを生成すべきかを判断するために使用できます。

表 6-3 で、制御ベクトルのビット位置を定義します。制御ベクトルのビット定義は、TIC の古いバージョンと下位互換性があるように設計されています。したがって、制御ビットのすべてが明白な位置にあるわけではありません。

表 6-3 制御ベクトルのビット定義

ビット位置	説明
0	制御ベクトル有効
1	予備
2	**HSIZE[0]**
3	**HSIZE[1]**
4	**HLOCK**
5	**HPROT[0]**
6	**HPROT[1]**
7	アドレスインクリメント・イネーブル
8	予備
9	**HPROT[2]**
10	**HPROT[3]**

TIC が実行できるバーストのタイプを制御するためのメカニズムはなく、不定長のインクリメント式バーストだけがサポートされています。TIC は 8 ビット、16 ビット、および 32 ビットの転送だけをサポートしています。したがって **HSIZE[2]** は変更できず、常に LOW です。

テストインターフェースを使用してバーストアクセスに対応するために、テストインターフェース・コントローラは、バスアドレスのインクリメントをサポートしています。TIC は 8 アドレスビットをインクリメントし、そしてこのインクリメンタがカバーできるアドレス範囲は、実行される転送のサイズに左右されます。

制御ベクトルは、TIC 内のアドレスインクリメンタを有効および無効にするメカニズムを提供します。これにより、内部 RAM のテストに使用される、インクリメンタルアドレスへのバーストアクセスが可能となります。一方、アドレスインクリメントは、バーストの連続アクセスが同じアドレスに対して行なわれるように無効にできます。これは、単一の周辺レジスタから連続的に読み出すために必要です。

HSIZE[1:0] がダイナミックに変化する場合、バーストモード・アクセスのためのアドレスインクリメンタのサポートは、バイト、ハーフワード、およびワード単位のインクリメントをサポートすることができなければなりません。そのため、適応型アドレスインクリメンタ・ロジックが必要です。

アドレスインクリメンタはデフォルトでは無効になっており、使用に先立って、制御ベクトルを使用して有効にしなければなりません。

Note

制御ベクトルは主に、アドレスバスと同じタイミングを持つ信号を変更するために使用されます。しかし、制御信号は、ロックされた転送が開始する前に実際に要求されるロック信号の変更も可能にします。テスト中に **HLOCK** 信号が使用される場合、その信号は、要求される前に転送に設定されなければなりません。**HLOCK** 信号に関するこのタイミングの相違により、実際にロックすべきシーケンスの前後に、追加の転送がロックされる場合があります。

6.6 AMBA AHB 試験手順の例

AHB 試験手順の例を、以下の見出しで説明します。

- *テストモードの開始*
- *書込みテストベクトル*：P.6-19
- *読出し転送*：P.6-20
- *制御ベクトル*：P.6-21
- *バーストベクトル*：P.6-22
- *読出しから書込みへ、および書込みから読出しへ*：P.6-23
- *テストモードの終了*：P.6-24

6.6.1 テストモードの開始

通常の動作モードでは、**TREQA** は LOW になり、テストアクセスが要求されていないことを示します。通常、外部バスインターフェースの一部となるテストバスが、必要に応じて通常動作に使用されます。テストモードを開始することにより、内部バスでの転送を発生させるテストベクトルを外部から適用できます。

テストモードを開始するためには、以下のシーケンスが必要とされ、これを図 6-4 に図解します。

1. テストバス・アクセスを要求するために、**TREQA** をアクティブにします。
2. TIC が内部バスの使用を認可されたときテストモードが開始します。これは、**TACK** 信号のアクティブ化によって示されます。
3. このポイントでは、**TCLK** が内部 **CHLK** 信号のソースになります。
4. テストモードが開始すると、アドレスベクトルを起動するために、**TREQB** がアクティブになります。
5. TIC は、有効なアドレスベクトルが適用されるまで、内部転送を行ないません。

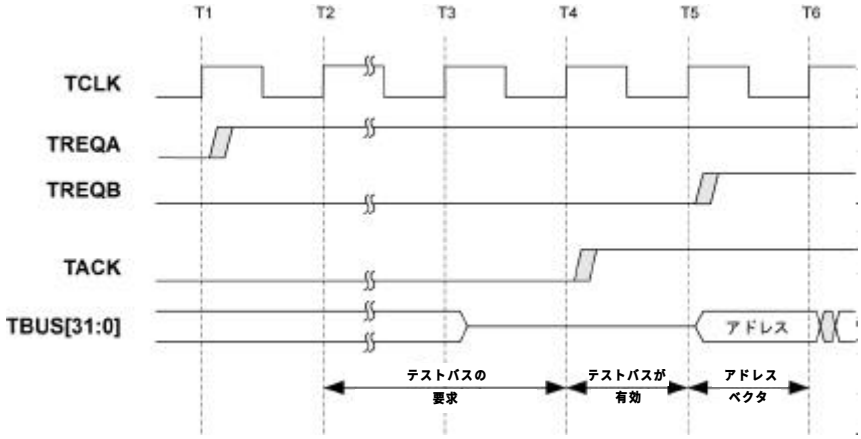


図 6-4 テスト開始シーケンス

同期テスターは、バスの TACK をポーリングするようになっていません

通常、TREQA は、バスへのアクセス権を確実に取得できる、できるだけ少ないサイクル数の間アクティブになります (最長のウェイトステートにある周辺ユニット・アクセスの終了、または現行命令を終了させるための全バス・マスターへの最大サイクル数のバスにおいて)。

6.6.2 書込みテストベクトル

図 6-5 には、一組の書込みテストベクトルを適用したときのイベントのシーケンスを示します。最初に、アドレスベクトルが適用され、これに書込みテストベクトルが続きます。

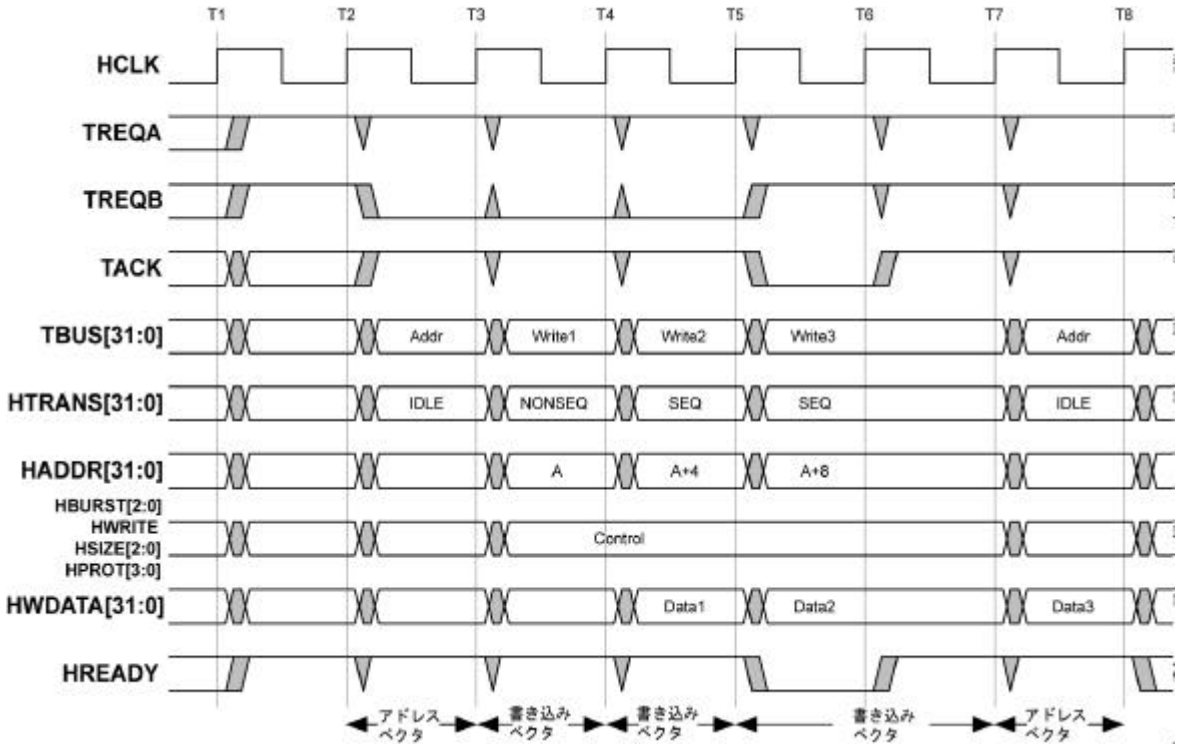


図 6-5 書込みテストベクトル

テストベクトルの書き込み時、以下の事項が適用されます。

- TREQA と TREQB 信号はパイプライン方式で、どの種類のベクトルが次のサイクルで適用されるかを示すために使用されます。図 6-5 に、実行されている多くの書込み転送の例を示します。
- TIC は時間 T3 でアドレスと TREQA/B 信号を取得します。これに続いて AHB 上の該当する転送を開始することができます。
- 次のサイクルでは、その書込みデータが TBUS 上に送られ、その後、次のクロックエッジ T4 で取得され、内部バス上に送られます。
- 内部転送が完了できない場合、TACK 信号が LOW になります。これは外部テストベクトルが別のサイクルの間に適用されなければならないことを示しています。

6.6.3 読出し転送

読出し転送は、TBUS を反対方向に駆動するように要求するので、もっと複雑です。したがって、TBUS の異なったドライバ間で切り換えるときのバスの衝突を防止するために、追加のサイクルが必要です。図 6-6 に、読出しに対する代表的なテストシーケンスを示します。

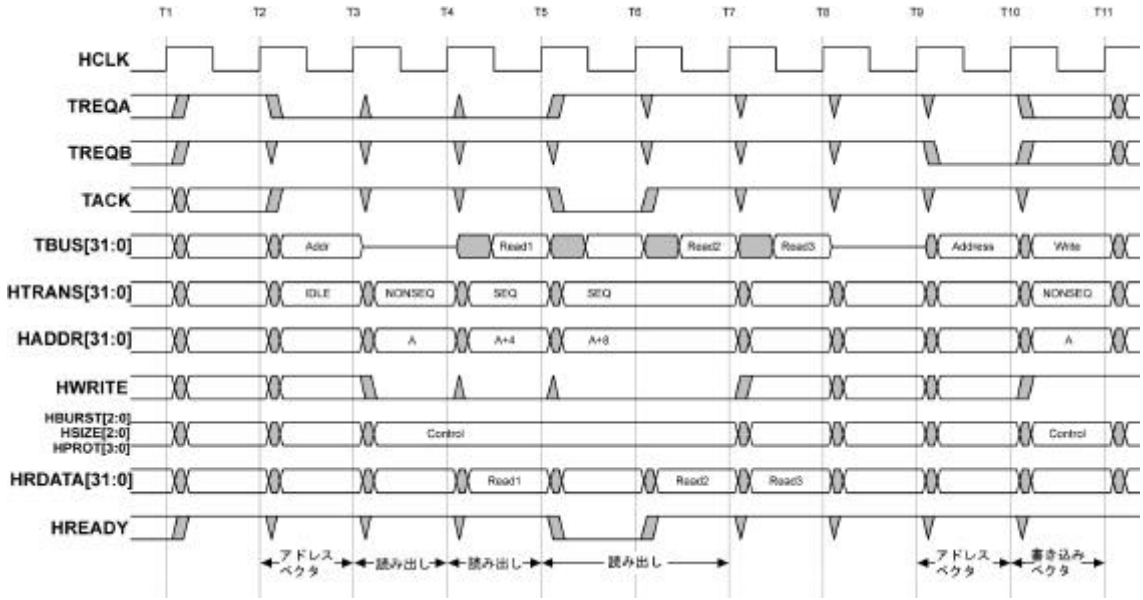


図 6-6 読出しテストベクトル

テストベクトルの読み出し時、以下の事項が適用されます。

- TREQA と TREQB 信号は、書込み転送の場合と同様に使用されます。最初に、アドレスベクトルを適用するために TREQA/B が使用され、後に続くサイクルでは、読出し転送が要求されていることを示すのに使用されます。読出しの最初のサイクルの間、TBUS はトライステート状態でなくてはなりません。これは、TIC がオンチップバッファによる読出しデータの送出を可能にする前に、そのバッファをトライステート状態にするように、TBUS を駆動する外部装置に 1 サイクル全体を当てられるのを保証します。
- 読出しバーストの終了時、バスターンアラウンドの時間を与えることも必要です。この場合、TIC は内部バッファをオフにすることが必要で、外部試験装置が駆動し始める前に 1 サイクル全体が与えられます。
- 読出しバーストの終了は、アドレスベクトルの場合、TREQA と TREQB の両方が HIGH になって示されます。実際、それらは 2 サイクルの間、アドレスベクトルを示す必要があります。これはバースト開始時のターンアラウンド・サイクルとバースト終了時のターンアラウンド・サイクルの両方を見込んでいます。

6.6.4 制御ベクトル

TIC の動作は、制御ベクトルを使用して修正することができます。2 つ以上のアドレスベクトルを連続して適用するときは、最後のベクトルが常に制御ベクトルと考えられ、アドレスとしてラッチされません。制御ベクトルのビット 0 は、その制御ベクトルが有効かどうか判断するために使用されます。これにより、制御情報を変更することなく複数のアドレスベクトルを適用できます。

図 6-7 には、制御ベクトルを挿入するプロセスを示します。

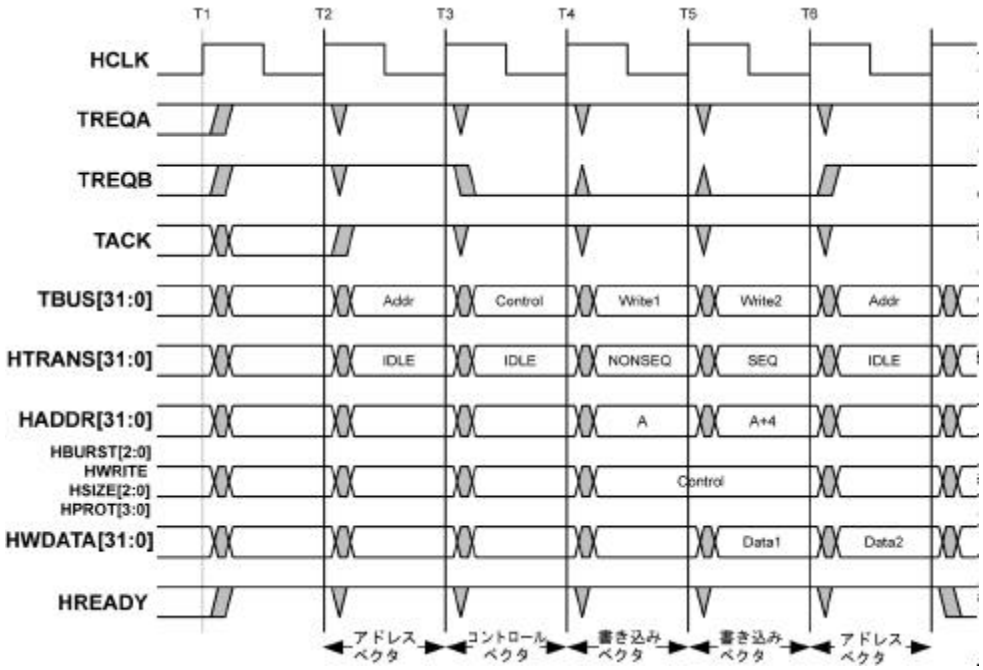


図 6-7 制御ベクトル

時間 T4 において、TIC は **TBUS** が制御ベクトルを含んでいると判断することができます。この理由は、前のサイクルがアドレスベクトルであった、かつ **TREQA/B** は次のサイクルが読出しか書込みのいずれかで、したがって現在のサイクルは制御ベクトルでなければならないということを示しているからです。

6.6.5 バーストベクトル

P.6-19 の図 6-5 と P.6-20 の図 6-6 における読出しおよび書込み転送の例は、バス上にバースト転送を形成するために追加の転送がどのように使用できるかを示しています。TIC はバースト転送の能力を制限しており、不定長インクリメント式バーストのみを実行することができます。

TIC は 8 ビットインクリメンタを組み込んでおり、インクリメンタ境界を越えるバーストを実行しようとすると、アドレスはラップし、TIC はその転送が NONSEQUENTIAL であるという信号を送信します。これが行なわれる正確な境界は、その転送のサイズに左右されれます。ワード転送の場合、インクリメンタは 1kB 境界でオーバーフローし、ハーフワード転送の場合は 512 バイト境界で、バイト転送の場合は 256 バイト境界でオーバーフローします。

6.6.6 読出しから書込みへ、および書込みから読出しへ

新しいアドレスベクトルを適用することなく、読出し転送と書込み転送の間での切り換えを可能にします。通常、これはアドレスインクリメンタを無効にした状態で行なわれます。したがって、読出し転送と書込み転送の両者とも、同じアドレスに対するものとなります。テスト環境上の必要に応じて、インクリメンタを有効にした状態でこれを行なうこともできます。

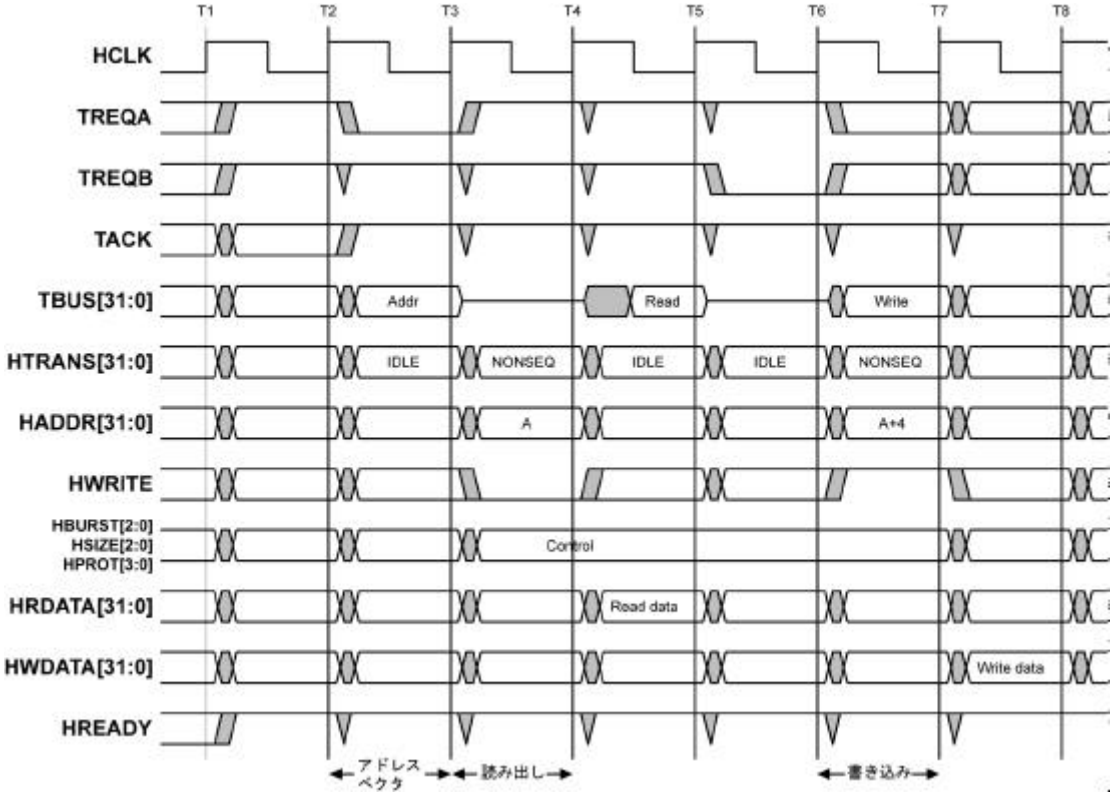


図 6-8 読出し転送、およびその後の書込み転送

読出し転送から書込み転送へ移行するときには、バスハンドオーバーに 2 サイクル見込むことも必要です。したがって、**TREQA** と **TREQB** は読出し後 2 サイクルの間、アドレスベクトルを送信すべきです。第三のアドレスベクトルを伴わない限り、これによってアドレスが変化することはありません。図 6-8 に、イベントのシーケンスを示します。

6.6.7 テストモードの終了

テストモードは、以下のシーケンスを使用して終了します。

1. 内部的に IDLE サイクルを発生する、1 サイクルのアドレスベクトルを適用します。これは、内部転送が完了したことを保証します。
2. **TREQA** と **TREQB** の両者が LOW になり、テストモードが終了することを示します。
3. テストインターフェースが通常のシステム動作に設定されたとき、**TACK** は LOW になり、テストモードが終了したことを示します。

TIC は量産テストだけでなく、システム動作中の診断テストにも使用出来るように、システムモードが手際良く開始し、終了可能となるのが重要です。

6.7 ASB テストインターフェース・コントローラ

図 6-9 に、ASB テストインターフェース・コントローラの状態図を示します。

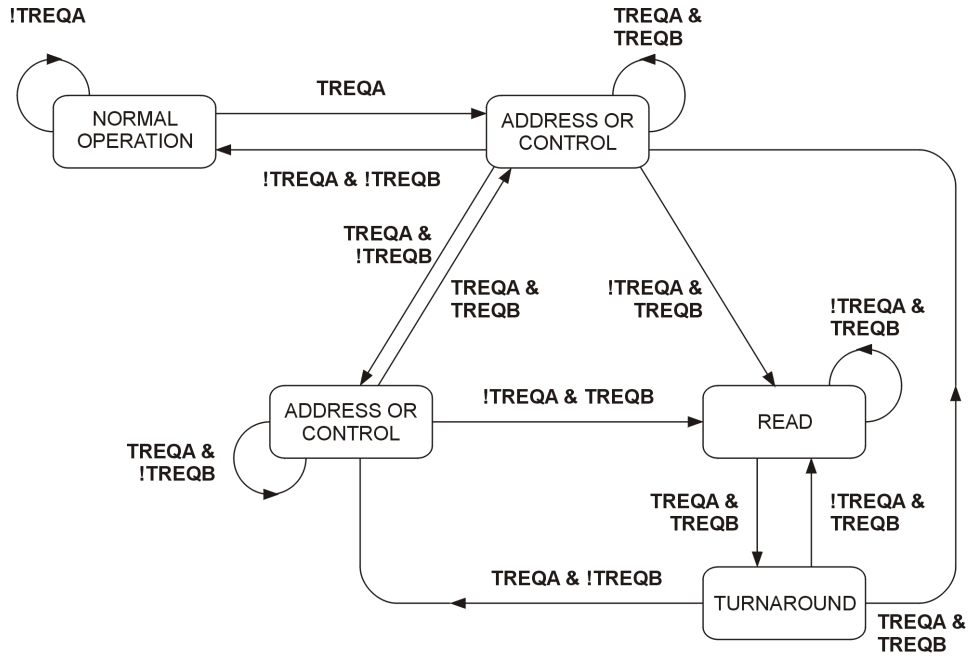


図 6-9 テストインターフェース・コントローラの状態図

TREQA と **TREQB** は、**TREQA** が ADDRESS OR CONTROL 状態への遷移に非同期的に使用される NORMAL OPERATION 以外の状態において、**TACK** が HIGH のときの **TCLK** の立下がりエッジで取得されます。リセット状態は、NORMAL OPERATION 状態です。

6.7.1 制御ベクトルのビット定義

実行できる転送のタイプを判断するために、TIC 内部に制御ベクトルが組み込まれています。この制御ベクトルは、**BSIZE**、**BPROT**、および **BLOK** の値を設定するためや、アドレスのインクリメントを制御するために使用されます。

制御パケットのバイト 0 は、内部システムバス上で行なわれるアクセスを規定するために使用されます。制御パケットのバイト 1 は、クロック制御とデバッグのために残されます。

表 6-4 に、制御ベクトルのビット割当てを示します。

表 6-4 制御ベクトルのビット定義

ビット位置	説明
0	制御ベクトル有効
1	予備
2	**BSIZE[0]**
3	**BSIZE[1]**
4	**BLOK**
5	**BPROT[0]**
6	**BPROT[1]**
7	アドレスインクリメント・イネーブル

6.8 AMBA ASB 試験手順の例

ASB 試験手順の例を、以下の項目に沿って説明します。

- *テストモードの開始*
- *アドレスベクトル*：P.6-28
- *制御ベクトル*：P.6-29
- *書込みテストベクトル*：P.6-31
- *バースト方向の変更*：P.6-36
- *テストモードの終了*：P.6-37

6.8.1 テストモードの開始

図 6-10 に示すように、テストモードは、以下のシーケンスを使用して開始します。

1. **TREQA** は、テストバス・アクセスを要求するために、アクティブになります。
2. テストモードは、TIC が内部バスの使用を認可されたときに開始します。これは、**TACK** 信号のアクティブ化によって示されます。
3. このポイントでは、**TCLK** が内部 **BCLK** 信号のソースになります。
4. テストモードが開始したとき、アドレスベクトルを起動するために **TREQB** がアクティブになります。

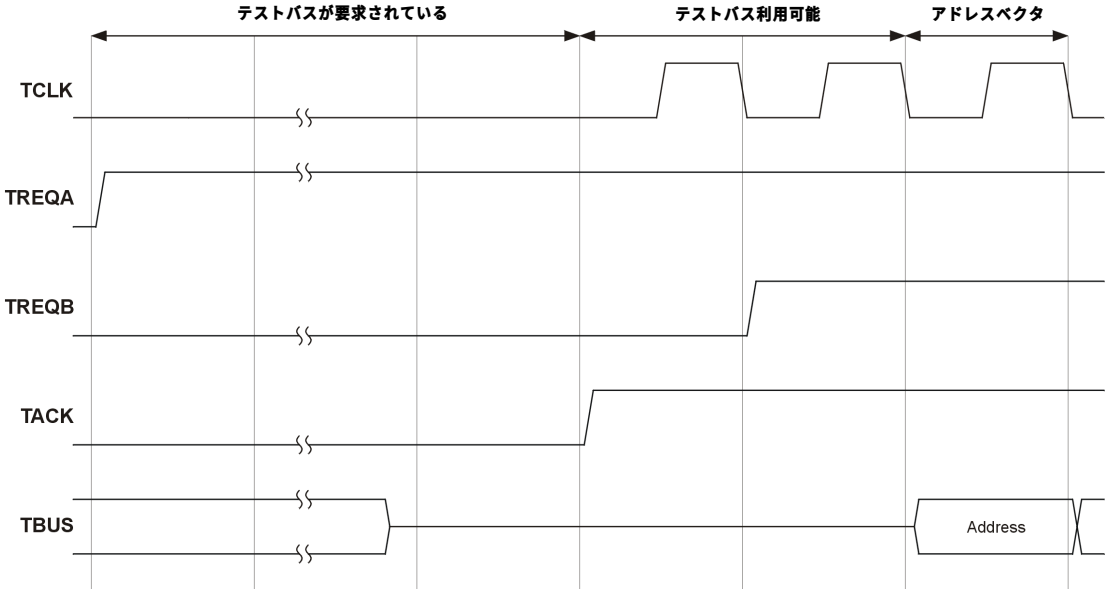


図 6-10 テスト開始シーケンス

6.8.2 アドレスベクトル

アドレスベクトルは、読出しまたは書き出し動作が行なわれる前に適用されなければなりません。図 6-11 に、書込みベクトルを伴う単一アドレスベクトルの例を示します。以下のシーケンスが行なわれます。

1. アドレスベクトルの次のサイクルを指示するために、**TREQA** と **TREQB** の両方が HIGH になります。
2. 後に続くテストベクトルのタイプを指示するために、**TREQA** と **TREQB** が変化している間にある次のサイクルで、アドレスが印加されます。
3. 次のサイクルで、書込み (または読出し) ベクトルが適用されます。

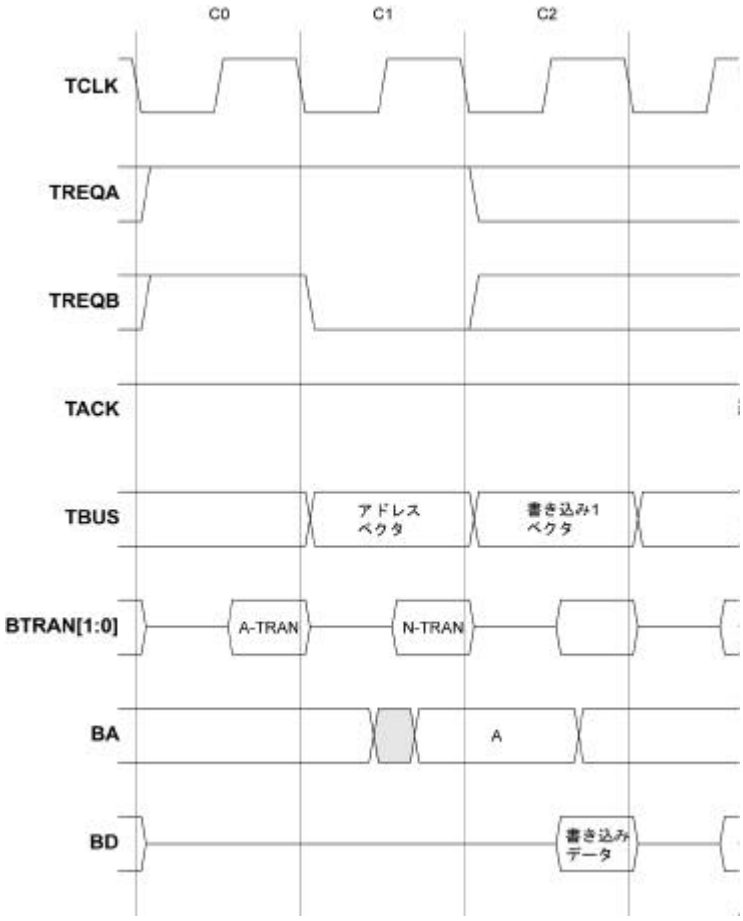


図 6-11 アドレスベクトル

6.8.3 制御ベクトル

制御ベクトルは常に、アドレスベクトルの後に続かなくてはなりません。図 6-12 に、書込みベクトルを伴うアドレスおよび制御ベクトルシーケンスを示します。以下のシーケンスが行なわれます。

1. アドレスベクトルが終了した後、制御ベクトルの次のサイクルを指示するために、**TREQA** と **TREQB** の両方が HIGH 状態に保持されます。
2. 後に続くテストベクトルの種類を反映するために **TREQA** と **TREQB** が変化している間にある、次のサイクルで制御情報が **TBUS[31:0]** に印加されます。このサイクル中、その制御ベクトルから影響を受けるすべての内部信号が変化します。

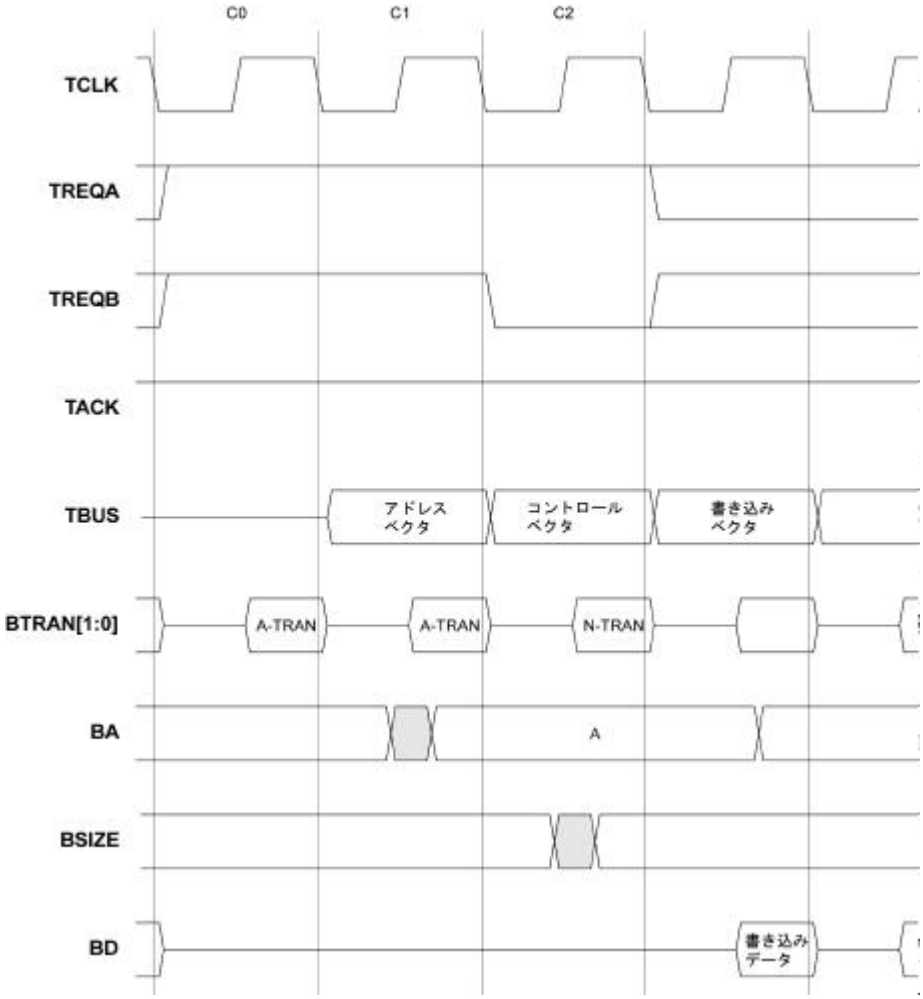


図 6-12 制御ベクトル

図 6-13 に、無効な制御ベクトルの後に続く転送の例を示します。TIC は、制御信号が変化しなかったため、内部バスで SEQUENTIAL 転送を行ないます。

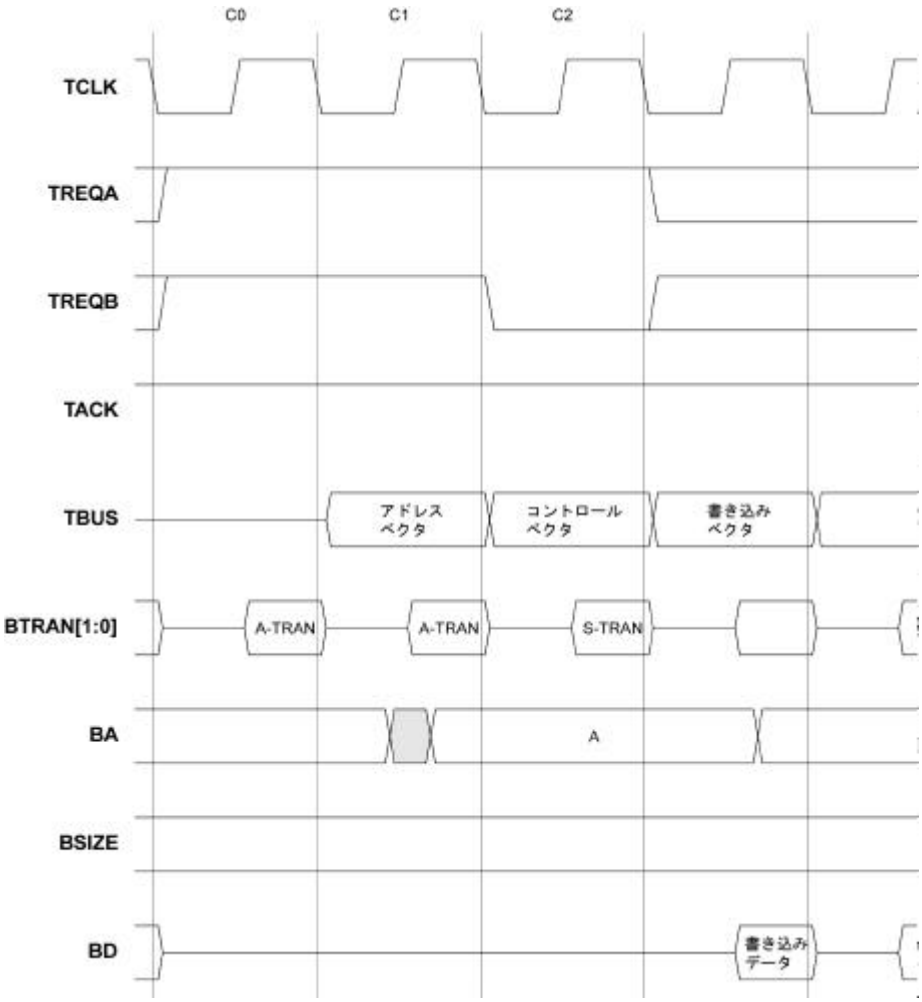


図 6-13 無効な制御ベクトル

6.8.4 書込みテストベクトル

図 6-14 に、単一アドレスベクトルの後に続く単一書込みベクトルの例を示します。

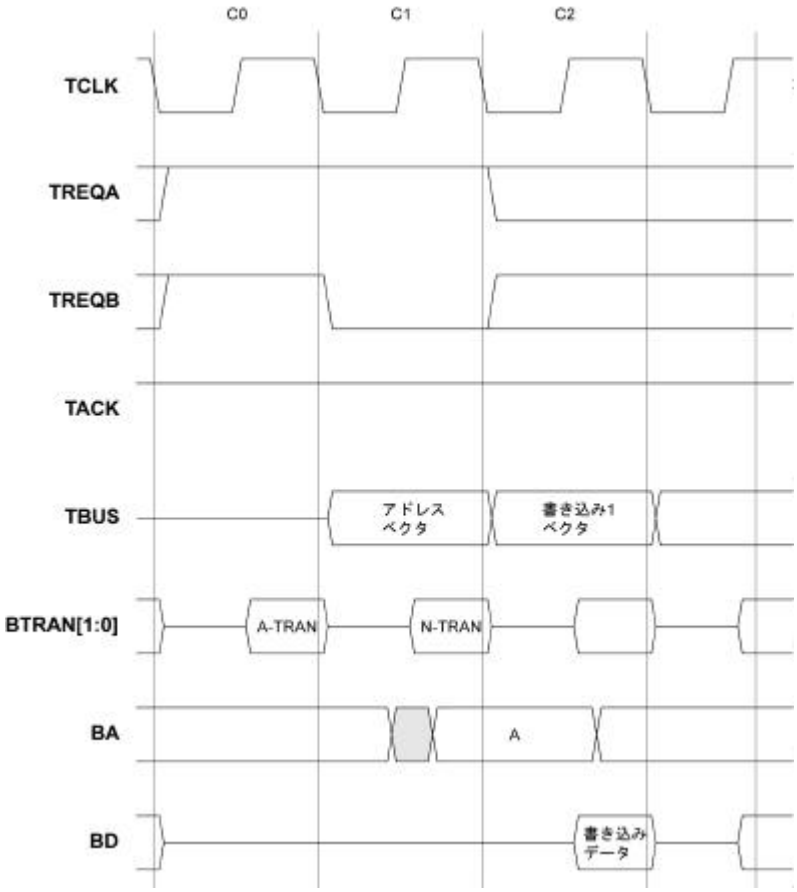


図 6-14 書込みテストベクトル

図 6-15 に、1 つのアドレスベクトルの後に続く延長書込みベクトルの例を示します。

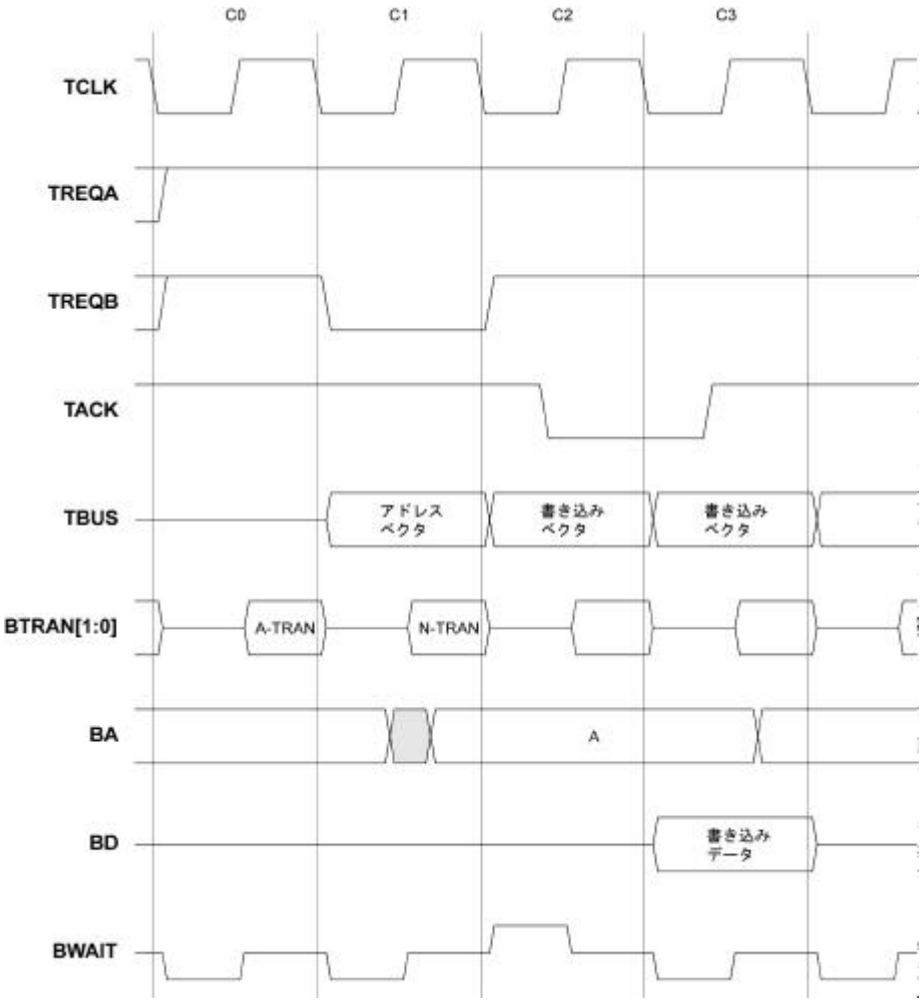


図 6-15 延長書込みテストベクトル

図 6-16 に、1 つのアドレスベクトル、続いて 1 つの読出しベクトル、そして最後に 1 つのターンアラウンド・ベクトルという、一連のベクトルの例を示します。

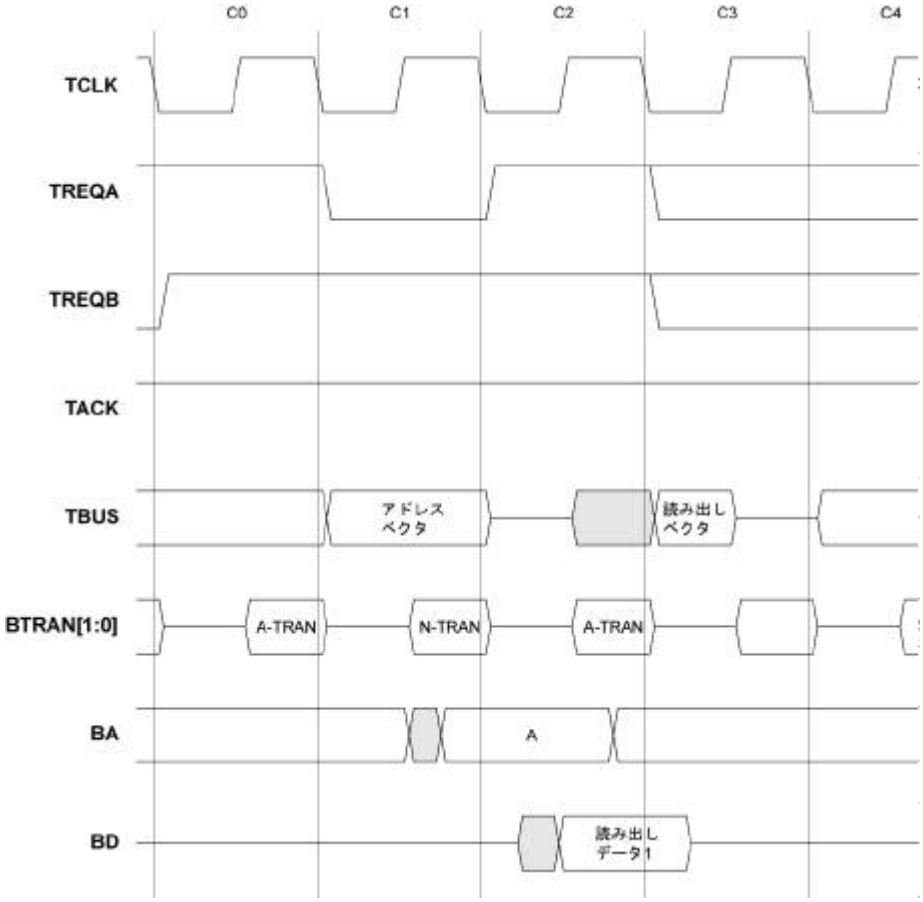


図 6-16 読出しテストベクトル

図 6-17 に、非インクリメント式アドレスへの SEQUENTIAL 転送を示します。

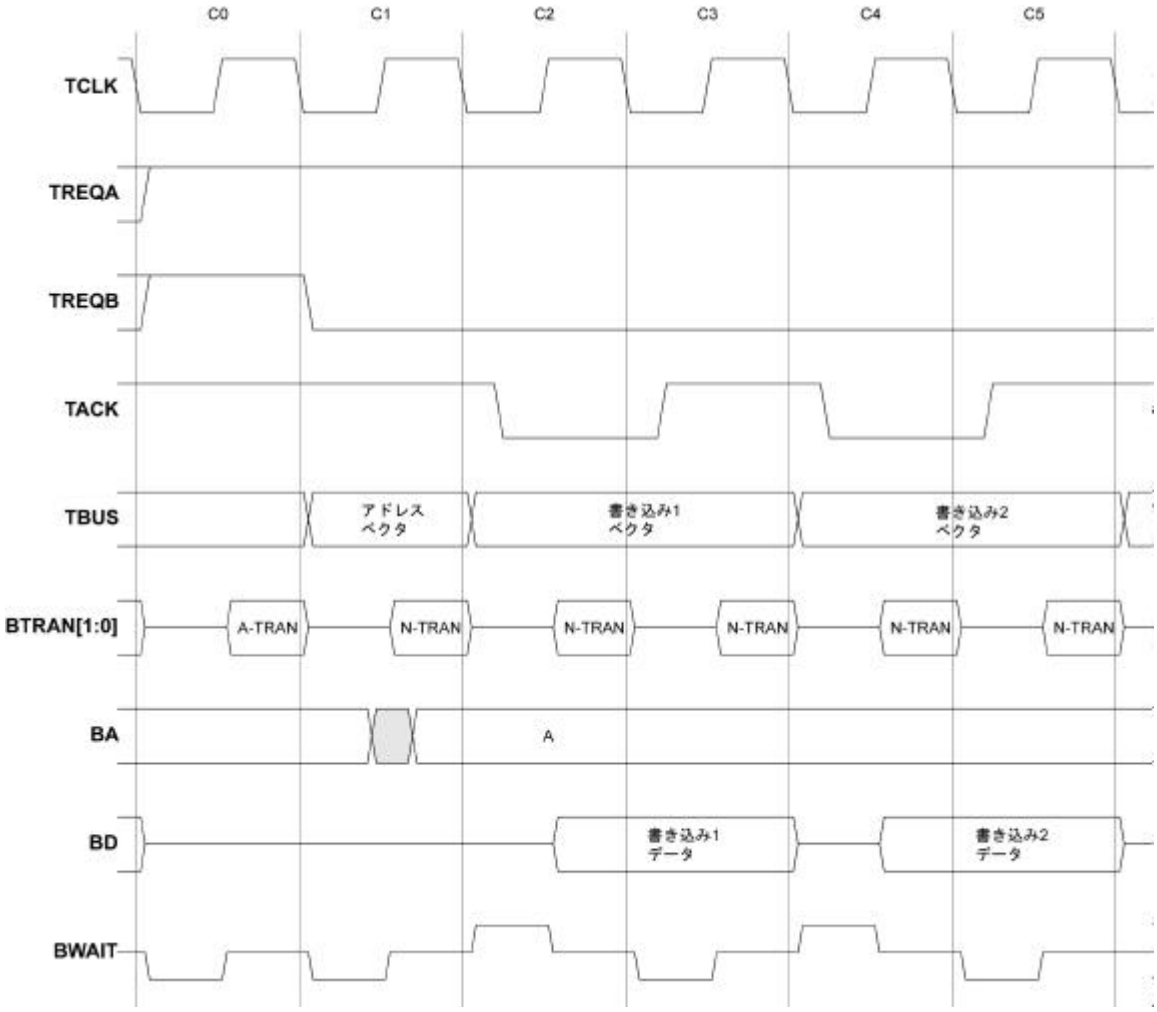


図 6-17 インクリメントを無効にしたバースト書込みベクトル

図 6-18 に、インクリメント式アドレスへの SEQUENTIAL 転送を示します。

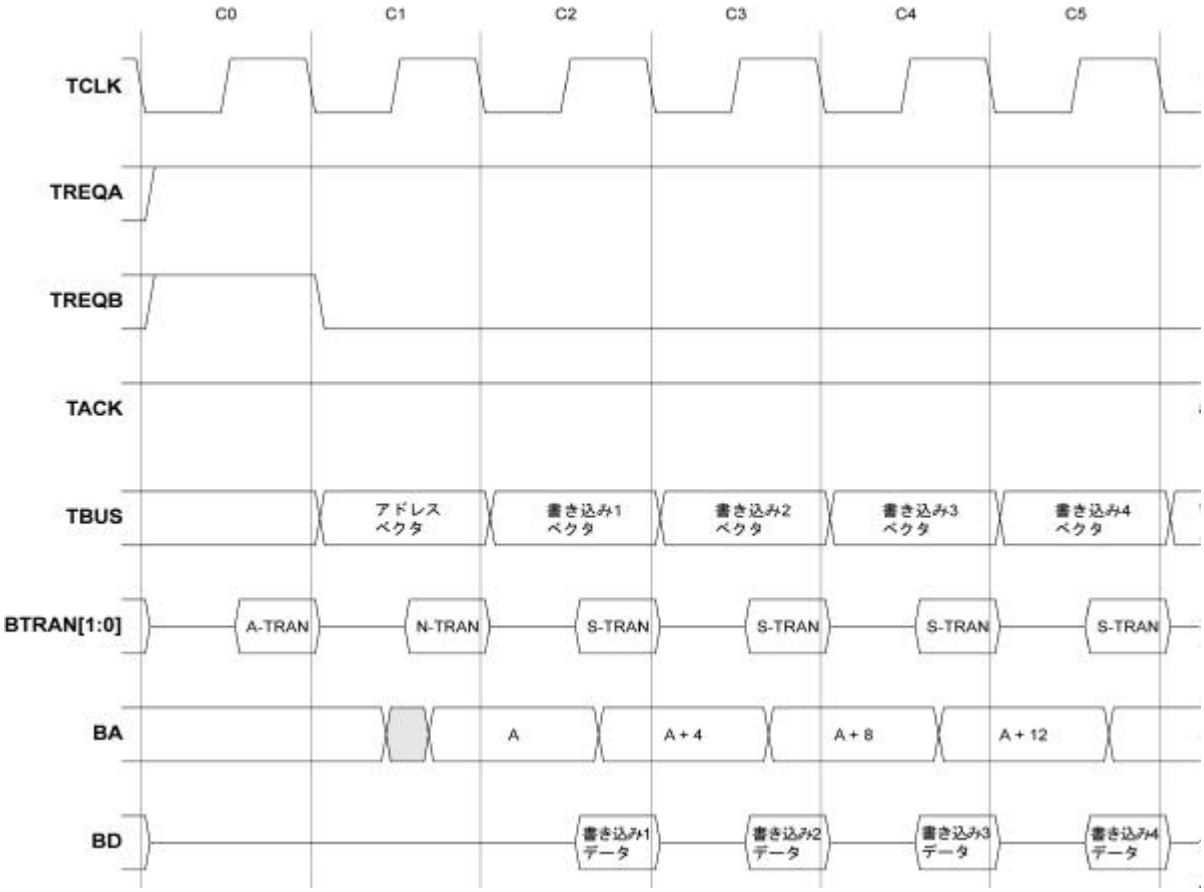


図 6-18 インクリメントを有効にしたバースト書込みベクトル

6.8.5 バースト方向の変更

図 6-19 に、方向が読出しから書込みへ変化するバーストを示します。.

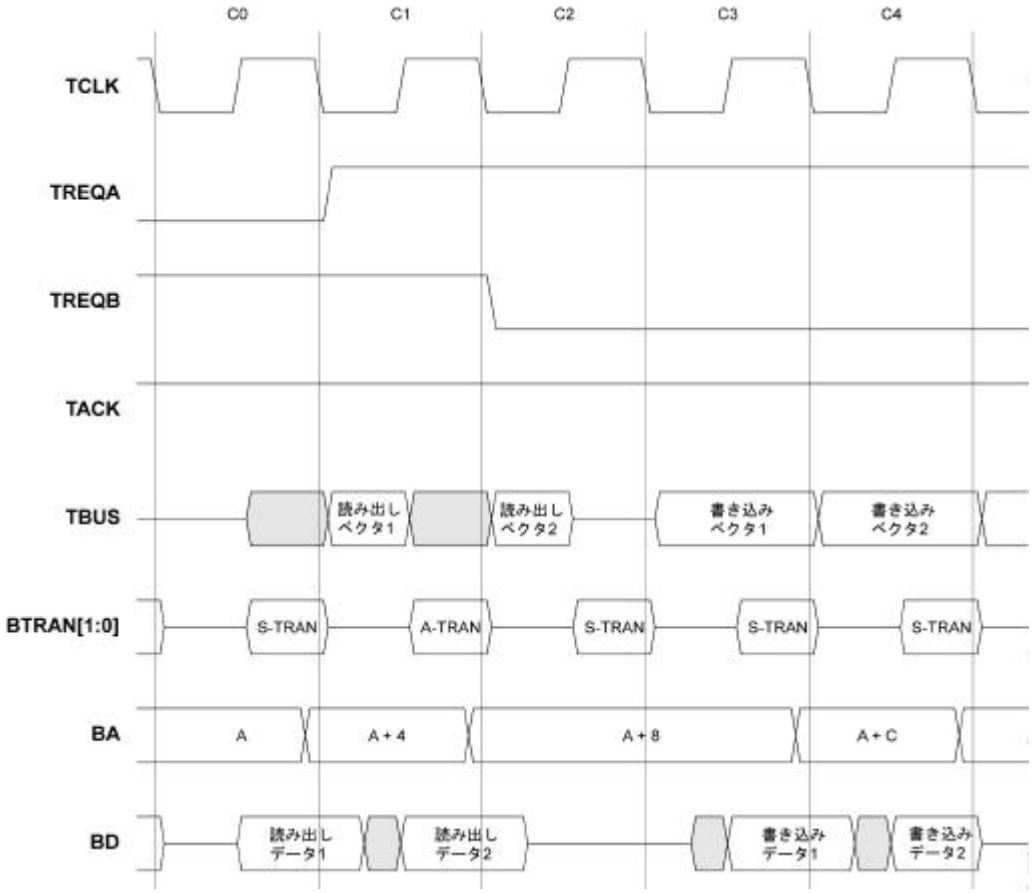


図 6-19 バースト方向の変更

6.8.6 テストモードの終了

図 6-20 に、テストモードの終了を示します。

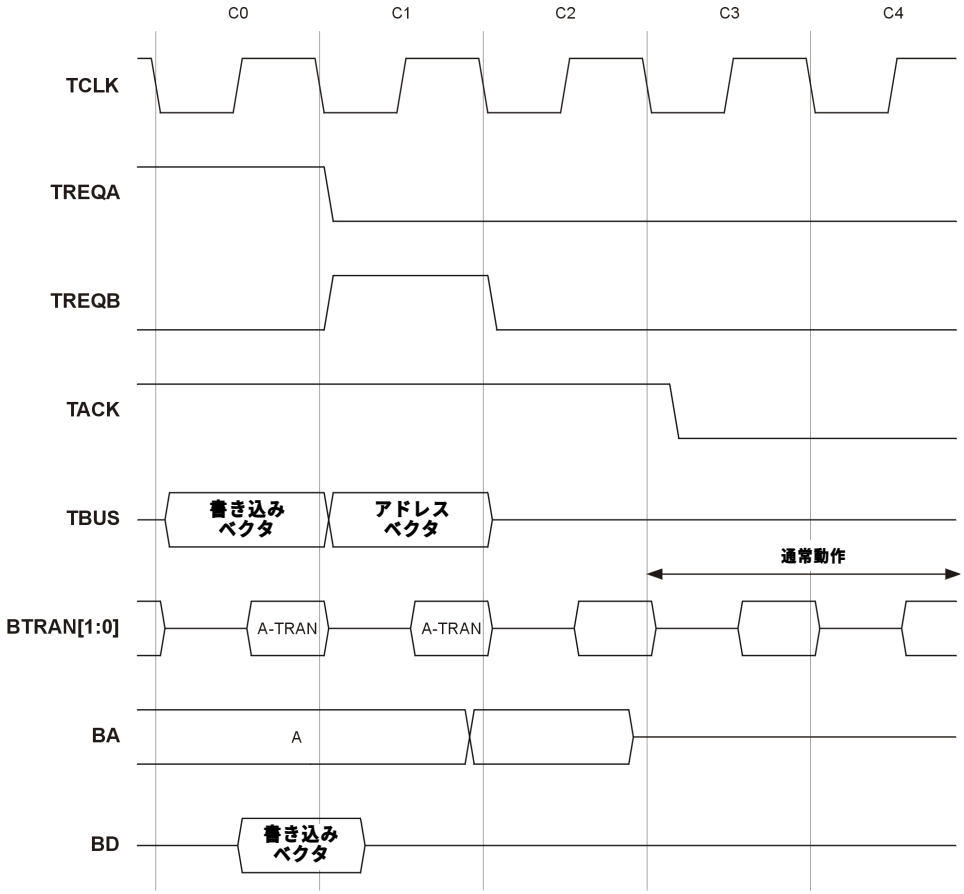


図 6-20 テストモードの終了

