



WP383 (v1.0) 2011 年 3 月 1 日

Virtex-7 および Kintex-7 FPGA で 高い DDR3 データ レートを実現

著者 : Adrian Cosoroaba

FPGA ベースのシステムでは、FPGA の内部メモリ容量を超えたデータを格納するために、外部メモリ インターフェイスが必要な場合があります。このメモリ インターフェイスには、FPGA とのデータ転送フローに十分対応できる読み書き帯域幅が必要となるため、システム全体のパフォーマンスに影響を与える可能性があります。

この I/O パフォーマンス要件とは、新しい世代の FPGA 製品ごとに、より高くなるメモリ インターフェイス帯域幅への要求です。高性能だけでなく、メモリ インターフェイス ソリューションには高い柔軟性と容易に実装できることが求められます。メモリ コントローラーやインターフェイス デザインはいろいろな意味で、システム全体の性能を決定するものです。

このホワイト ペーパーでは、さまざまなメモリ インターフェイスとコントローラー デザインでの課題、および Virtex®-7 と Kintex™-7 FPGA で 1.866 Gb/s の DDR3 データ レートを実現する 7 シリーズ FPGA の高性能ソリューションについて説明します。

メモリ インターフェイスの傾向とザイリンクス ソリューション

ダブルデータレート (DDR) SDRAM の世代が進むに連れて、より高いシステム性能の要件を満たすためにピンのデータレートも高くなりました。

FPGA ベースのメモリ コントローラーの実装では、このような傾向に対応するため、さまざまな課題に取り組んできました。高データレート化に伴う課題には、インターフェイスとコントローラーの実装において2つの重要な面があります。

- FPGA 物理層 (PHY) における読み出しデータの取得
- FPGA ロジック ファブリックにおけるコントローラーの実装

読み出しデータの取得

データレートが増加すると、読み出しデータの有効ウィンドウが小さくなり、タイミング マージンが減少するため、読み出しデータの取得が難しくなります。FPGA I/O ブロックとクロッキング エLEMENTは正確なキャプチャ メカニズムを提供する必要があり、電圧と温度が変動する環境でもデータ キャプチャ クロックがデータ有効ウィンドウ内に維持されなくてはなりません (図 1 参照)。

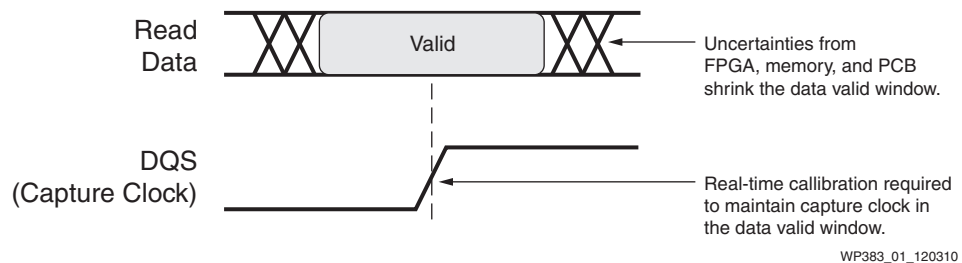
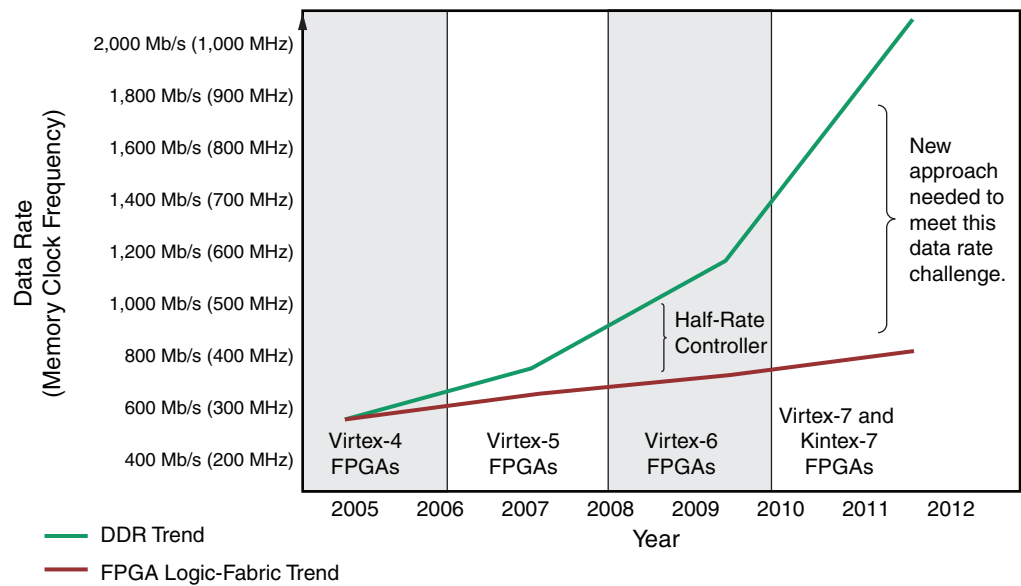


図 1: データ クロックおよびリアルタイム キャリブレーションの課題

つまり、データレートの増加とタイミング マージンの縮小により、さらに正確なクロッキングとキャリブレーション回路が要求されるようになります。このような回路は、変動するシステム条件下でも十分なセットアップ/ホールド マージンを保ちながら、キャプチャ クロック (DQS) がデータ有効ウィンドウ内に維持されているように、堅牢なリアルタイム キャリブレーション メカニズムを提供する必要があります。7 シリーズ FPGA には Phaser と呼ばれる新しいクロッキング アーキテクチャ エLEMENTがあり、Virtex-7 および Kintex-7 FPGA では1.866Gb/s データレートのリアルタイム キャリブレーションがサポートされます。

コントローラーの実装

読み出しデータの取得と同様に重要な要素が、メモリ コントローラーのロジック ファブリック ベースの実装です。DDR メモリ データレートの増加 (メモリ クロック レートの 2 倍) に伴い、メモリ コマンドおよびデータ フローを管理する FPGA ロジック ファブリック ベースのコントローラーも高いレートで動作することが求められます。FPGA の世代が進むにつれて、DDR アーキテクチャでのメモリ クロック レートはロジック ファブリックの性能よりも速いペースで増加しているため、求められるメモリ クロック レートとロジック ファブリック ベースのコントローラーにおけるクロック レート間にギャップが生じるようになりました (図 2 参照)。このギャップを埋めるために、ザイリンクスは革新的なソリューションを開発しました。



WP383_02_122010

図 2: FPGA ロジック ファブリック ベースのコントローラーでのデータ レート 課題

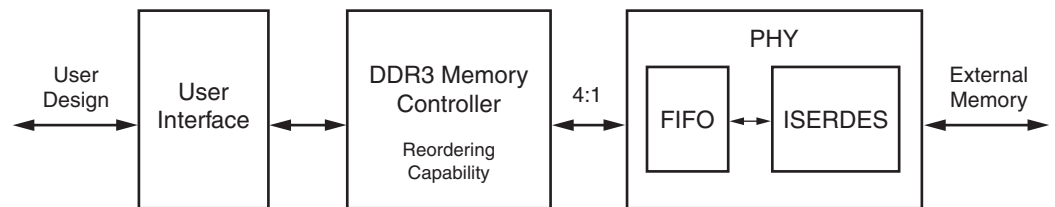
Virtex-6 FPGA などの前世代製品では、ロジック ファブリックのタイミング制限を満たすためにメモリ クロックの半分のレートで動作する DDR3 コントローラーが実装されています。このような実装では、I/O のデータ レートとコントローラーのために求められるロジック ファブリックのクロック レートがバランス良く作用しています。

一方、7 シリーズ FPGA では、最大 1.866Gb/s の DDR3 データ レート (高性能アプリケーションで必要) とロジック ファブリックのクロック レート間のギャップを新しいアーキテクチャが補います。

7 シリーズ FPGA のメモリ インターフェイス アーキテクチャ

ザイリックスは、7 シリーズ FPGA で PHY 層メモリ インターフェイスおよびコントローラーのアーキテクチャを改善して 1.866Gb/s のデータ レートを実現すると共に、DDR3 コントローラー デザインの効率性も改善して、より高い帯域幅を可能にしました。

図 3 に示すように、メモリ コントローラーとインターフェイスのアーキテクチャには、3 つの機能モジュール (PHY、DDR3 メモリ コントローラー、ユーザー インターフェイス) が統合されています。



WP383_03_120610

図 3: DDR3 コントローラーとインターフェイスのアーキテクチャ

1.866Gb/s データ レートで動作する DDR3 インターフェイスの場合、933MHz の DDR3 メモリ クロック レートに対応できるコントローラー ステート マシンをロジック ファブリックに実装するために、ロジック ファブリック ベースのコントローラーにはメモリの 1/4 のクロック レートが必要です。改善された PHY アーキテクチャでは、専用の FIFO を使用してこれを可能にします。この FIFO は、ロジック ファブリック ベース コントローラーのクロック レートからメモリ クロック レートを適切な比率で分離するギアボックス機能を提供します。この場合の比率は 4:1 となりますが、メモリ クロック レートが低い場合は 2:1 となります。800MHz (1,600Mb/s) 以上のクロック レートが必要な DDR3

インターフェイスには 4:1 が適しており、550MHz 未満のレートで動作する QDR-II+ あるいは RLLDRAM II などのインターフェイスには 2:1 が適しています。

物理層の最適化

PHY は、読み出しデータの取得やコントローラーから外部メモリ デバイスへ書き込みデータを転送する際に使用されます。I/O ドライバーは、最高スピード グレードの DDR3 デバイスでサポートされる最大 1.866Gb/s のデータ レートに対応する、十分な駆動能力とスイッチング スピードを提供する必要があります。革新的な I/O アーキテクチャでは、このような高いレートでも優れたシグナル インテグリティを実現する駆動能力が提供されています。1.866Gb/s に対応する駆動要件を満たすのに十分な信号ゲインを確保するには、2V プリドライバ電圧オプション (V_{CCAUX_IO}) が必要でした (図 4 参照)。

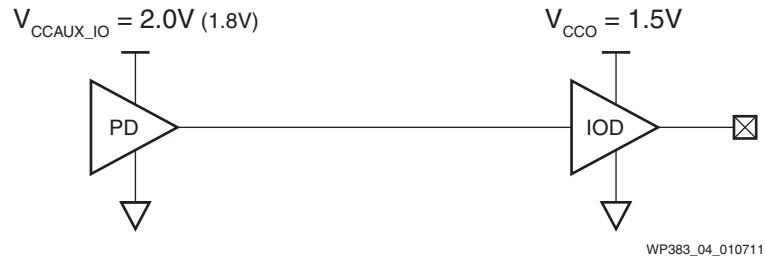


図 4: 2.0V プリドライバ (V_{CCAUX_IO}) 電圧供給オプション

高データ レートで I/O ドライバやレシーバをスイッチングする機能を可能にするだけでなく、読み出しデータの取得やリアルタイム キャリブレーションには、データ有効ウィンドウのクロックの初期キャリブレーションを管理でき、またシステム動作中に電圧や温度が変動してもこの関係を維持できる専用のクロッキング回路が必要です。7 シリーズ FPGA には、クロック マネジメント タイル (CMT) の一部として、最高 7ps の精度でクロックとデータ読み出しのタイミングを制御し、維持する機能をすべて備えた Phaser と呼ばれる新しいクロッキング構造が統合されています。

図 5 に、PHY アーキテクチャの基礎を構成する I/O ブロック、および CMT とロジック間の信号関係を示します。システムクロックが CMT 内の位相ロックループ (PLL) を駆動し、次に BUFG を介してメモリコントローラーのクロックを駆動します。PLL 出力は、PHY コントロールや PHASER_IN/PHASER_OUT ブロックも駆動します。

PHASER_IN ブロックでは、主に 2 つのファンクションを実行します。1 つは、ISERDES でデータ (図 5 の DQ) をキャプチャするために使用される DQS 信号に遅延を与えます。もう 1 つは、ISERDES から IN_FIFO ブロックへのデータ転送を制御します。

FIFO には 1:2 および 1:1 の 2 つのモードがあります。1:2 モードは、より低いレートで動作する必要があるロジックファブリックから高いデータレートの I/O を分離します。1:4 ISERDES および 1:2 FIFO の転送率の場合、データはメモリクロックレートの 1/4 のシングルデータレートで効率的に送信されます。これは、1,600Mb/s (または 800MHz) より高いデータレートの場合に必要なです。メモリコントローラーに必要なレートは、800MHz の 1/4 です。メモリとコントローラーの 2 つのクロックシステムを 1:4 で分離すると、タイミングクロージャが容易に達成できるというメリットがあります。1.866Gb/s のデータレート (933MHz のクロックレート) の場合、メモリコントローラーは 233MHz で動作します。クロックレートがこの範囲であれば、ロジックファブリックベースのコントローラーステートマシンは、タイミングを満たすことができます。

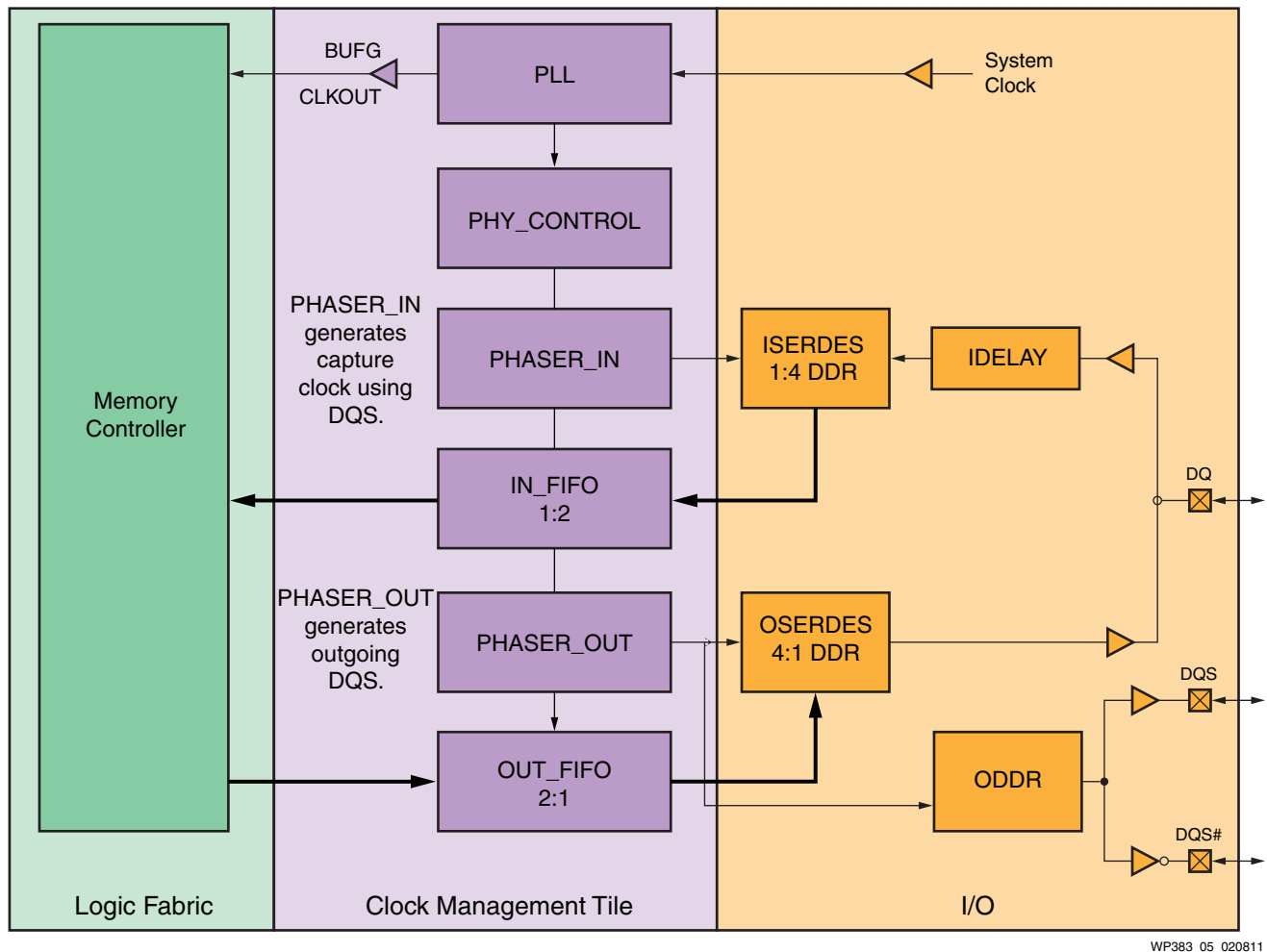


図 5: PHY アーキテクチャとロジック ファブリック ベースのメモリ コントローラーへのデータ転送

出力のデータパスとクロッキングは入力パスと類似していますが、PHASER_OUT と OUT_FIFO を使用してクロックを供給し、データ出力転送を制御します。PHASER_OUT は出力 DQS 信号の生成も制御します。

DDR3 SDRAM コントローラーでさらに高性能を実現

ピンでの最大データレートは、メモリ インターフェイスの性能を決定する 1 つの要因にすぎません。全体的な性能を決定する実際のあるいは平均帯域幅が、ピンのデータレート、効率性、バス幅の要因となります。

DDR3 コントローラーのリオーダーリング (順序変更) 機能を有効にすることで、より高い性能が実現されます。読み出しをリオーダーリングして読み出しと書き込みをグループ化することで、データバスを効率的に使用します。効率性は、データを転送するクロック サイクル数を合計クロック サイクル数で割った値として定義できます。

$$\text{Efficiency (\%)} = \frac{\text{Clock cycles transferring data}}{\text{Total clock cycles}} \times 100 \quad \text{数式 1}$$

単にクロック周波数を増加させるだけでは、より高い帯域幅を得ることはできない場合があるため、効率性は非常に重要です。クロック周波数の高いシステムを実装した場合、コストがかかり、消費電力が増加する可能性があります。あるクロック周波数で実現可能な最大帯域幅を利用できないということは、最適な設計手法ではありません。システムによっては、コントローラーがデータバスを効率的に使用していないと、実際の帯域幅は最大よりも 10% 低下する可能性があります。バスに RWRW コマン

ドが発行されると、ターンアラウンド タイム (応答時間) に多くのデッド サイクル (間隔) が生じます。さらに、読み出しまたは書き込みコマンドが既にオープン行へアクセスしていない場合は、プリチャージ タイムがデッド サイクルに追加されることになります。DDR3 SDRAM の動作にはこのようなデッド サイクルが生じるものであり、リオーダーリング コントローラーでこれを軽減できます。

最大帯域幅を実現するには、コントローラーが RWRW による DDR3 SDRAM へのアクセスを最小限にし、オープン行へのアクセスを最大限にする必要があります。ザイリンクスの 7 シリーズ FPGA DDR3 コントローラーは、量産システムで実績のある Virtex-6 FPGA DDR3 コントローラーのリオーダーリング アーキテクチャを利用しています。コントローラーは、ユーザー コマンドをリオーダーリングするための追加情報を処理するコマンド キューを使用して、次のことを可能にします。

- 読み出しと書き込みをグループ化することによって RWRW アクセスの切り替えをリオーダーリングし、バスのターンアラウンド タイムを最小化する
- ページミスが最小限になるように読み出しをリオーダーリングし、より多くのコマンドがオープン行へアクセスできるようにする

また、リオーダーリング コントローラー モードをオフにするオプションもあるため、必要に応じて順序変更なしでも実装できます。

リオーダーリング DDR3 コントローラーのテスト

効率性の改善を評価するため、Virtex-6 FPGA の DDR3 コントローラーにいくつかのテスト パターンをインプリメントし、DDR3 コントローラーでリオーダーリング アルゴリズムを使用しない等価デバイスと比較しました。

図 6 から、ランダムおよび交互の読み出し/書き込みパターンがデータ バスの効率性を低下させていることがわかります。また、ザイリンクスの DDR3 リオーダーリング コントローラーでは、リオーダーしないコントローラーを実装した場合よりも帯域幅が平均して 2 倍向上していることも示されています。ベンチマークで使用されている DDR3 リオーダーリング コントローラーには最大 8 個のコマンド列 (バンク マシン) があります。7 シリーズ FPGA は、Virtex-6 FPGA と同じ DDR3 リオーダーリング コントローラー オプションを備えています。

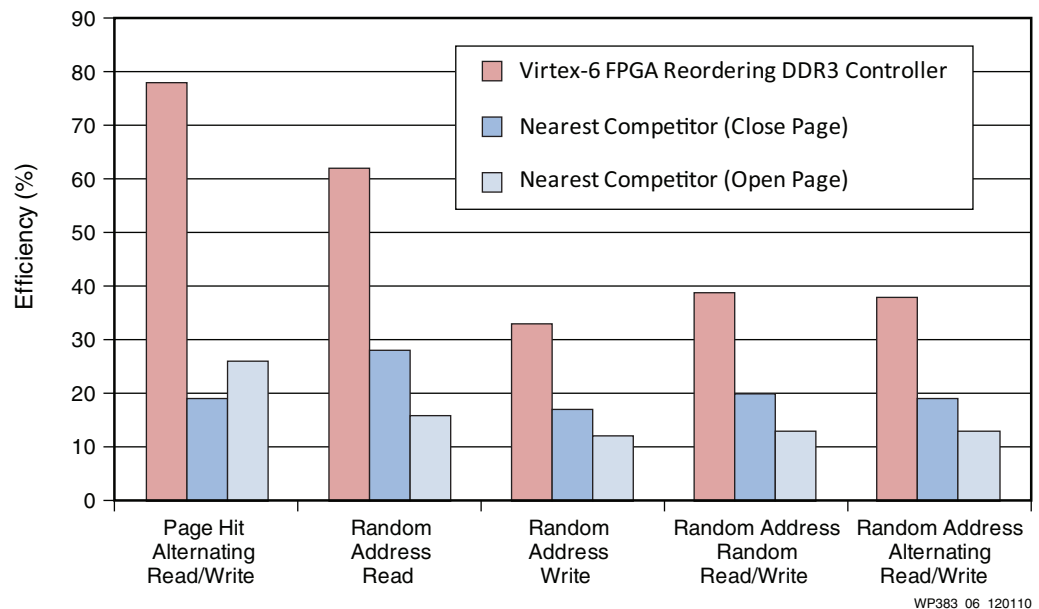
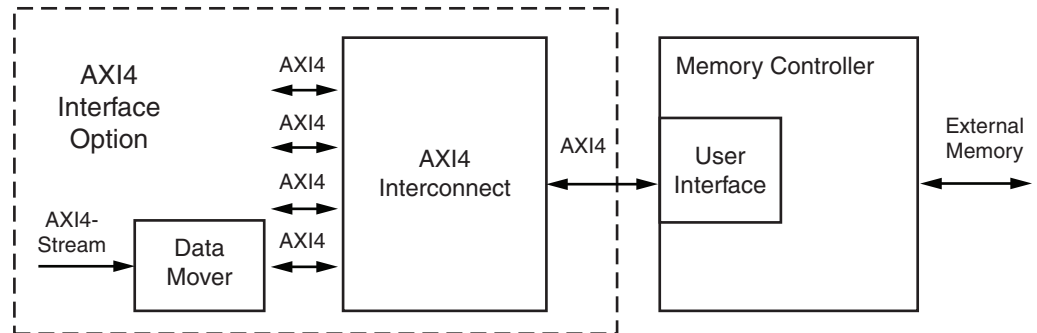


図 6: ザイリンクスのリオーダーリング コントローラーを使用した場合の帯域幅の効率性と向上率

より高い性能と柔軟性を可能にするユーザー インターフェイス オプション

メモリ コントローラーへのインターフェイス接続は、メモリ コントローラー クロックと同期して動作するユーザー インターフェイスを介して行われます。7 シリーズ FPGA では、さまざまなアプリケーションに対応できる高い柔軟性と性能を可能にするため、2 つのユーザー インターフェイス オプションを提供しています。(図 7 参照)。



WP383_07_121410

図 7: 7 シリーズ FPGA でのメモリ コントローラー ユーザー インターフェイス オプション

外部メモリ インターフェイスがシステムの性能を決定づけるアプリケーションでは、一般的にシングル ユーザー ポートが使用されます。7 シリーズ FPGA で提供されているオプションの 1 つが、Virtex-6 FPGA インプリメンテーションと同様に、コントローラーと同一クロック レートで動作するシンプルな FIFO ベースのユーザー インターフェイスです。

2 つ目のオプションは Advanced Microcontroller Bus Architecture (AMBA®4) AXI4 であり、ビデオまたはエンベデッド アプリケーションなど、マルチポートや AXI4 が要求されるアプリケーションで使用されます。AXI4 マルチポート機能によって、ユーザーは AXI4 プロトコルを介してメモリ コントローラーを共有できます。AXI4 インターコネクト ブロックは、複数ポート (ユーザー) がコントローラーへアクセスしたり、ほかの AXI4 IP ヘシームレスに接続するための接続性を提供します。さらに、データムーバーブロックは、AXI4-Stream を使用して接続する際の追加機能を提供します。この AXI4 準拠のオプションでの性能は、シンプルなシングル ポート ユーザー インターフェイスと同程度ですが、以前のマルチポート メモリ コントローラー (MPMC) インプリメンテーションと比較すると、エンベデッド アプリケーションではるかに高い性能が達成できます。AXI4 インターコネクトの利点については、[WP379](#) :『プラグアンドプレイ IP への道を開く AXI4 インターコネクト』を参照してください。

メモリ インターフェイス ジェネレーター (MIG) で高い生産性を実現

メモリ コントローラーおよびインターフェイスのデザインは、MIG のウィザードを使用して生成できます。MIG ソフトウェアは、CORE Generator™ ツールの一部としてザイリンクスより無償で提供されています。MIG デザイン フローは、従来の FPGA デザイン フローと類似しています。MIG の利点は、ユーザーが PHY インターフェイスやメモリ コントローラー用の RTL コードを最初から生成する必要がないことです。MIG ウィザードが RTL (HDL コード) と UCF ファイル (制約ファイル) をそれぞれ生成します。これらのファイルは、ハードウェアで検証された IP ライブラリに基づいており、ユーザー入力が反映されます。MIG は、高い性能を実現するために必要な制約を適用して、あらかじめ検証済みの IP を生成します。

生成された RTL コードや制約ファイルはユーザーによる変更が可能です。ブラックボックスが含まれるその他のソリューションとは異なり、MIG ツールは暗号化されていないコードを出力するため、デザインを変更してさらにカスタマイズできる柔軟性があります。

MIG の出力ファイルは、デザインの異なる構築ブロックに適用するモジュール別 (ユーザー インターフェイス、PHY、コントローラー ステート マシンなど) に分かれています。オプションとして、MIG で生成された PHY へ異なるコントローラーを接続することも可能です。さらに、MIG はメモリ チェッカー機能付きの合成可能なテストベンチを生成します。テストベンチは、ファンクション シミュレー

ションやザイリンクス ベース デザインのハードウェア検証で使用されるデザイン例であり、メモリ コントローラーに対して一連の書き込みおよび読み出し命令を発行します。また、カスタム テストベンチの生成、帯域幅の効率性予測、さまざまなメモリ アクセス パターンの予想性能を検証するためのテンプレートとしても使用できます。

より高い性能を実現するためのハードウェア検証および特性評価

7 シリーズ FPGA 用のメモリ インターフェイスとコントローラー IP のハードウェア検証は、堅牢かつ信頼性の高い高性能ソリューションを提供するために重要な最終ステップです。ザイリンクスは、実際のシステム テスト プロセスで特性評価手法を使用し、複数の世代の FPGA をターゲットとしてメモリ インターフェイス デザインを検証および特性評価してきました。特性評価プロセスは、実際のユーザー システムと同等あるいはそれを超える厳しいシステム環境でプロセス/電圧/温度 (PVT) の限界値 が確実に評価されるように、さまざまなテストに基づいて行われています。

まとめ

メモリ インターフェイスでより高い性能を達成するには、高い最大データ レートを実現するシリコンの機能と、そのような高いデータ レートを維持できる効率に優れたメモリ コントローラーを使用して設計することです。ザイリンクスは、7 シリーズ FPGA で、要件の厳しいアプリケーション向けに 1.866 Gb/s の DDR3 データ レート性能を実現し、この高データ レートを維持するようにメモリ コントローラーの効率を向上させることによって、業界で最高性能を誇るメモリ インターフェイス ソリューションを開発しました。また、コア IP の迅速なカスタマイズをサポートし、簡単に使える MIG のようなソフトウェア ツールを継続して提供しています。ザイリンクスのメモリ インターフェイス ソリューションは、広範囲のハードウェア特性評価に基づいており、電圧および温度が変動する条件下の標準的システムで確実に高性能を実現するものです。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2011/03/01	1.0	初版リリース

Notice of Disclaimer

The information disclosed to you hereunder (the “Information”) is provided “AS-IS” with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

本資料は英語版 (v1.0) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

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