



XAPP858 (v1.1) 2007 年 1 月 9 日

## Virtex-5 デバイスの高性能 DDR2 SDRAM インターフェイス

著者 : Karthi Palanisamy, Maria George

本資料は英語版 (v1.1) を翻訳したものです。英語の更新バージョンがリリースされている場合には、最新の英語版を必ずご参照ください。

### 概要

このアプリケーション ノートでは、高性能 DDR2 SDRAM インターフェイスのコントローラおよびデータ キャプチャ方法について説明します。この方法では、Virtex™-5 の全 I/O で利用可能な ISERDES (Input Serializer/Deserializer) および ODDR (Output Double Data Rate) 機能を使用します。

### はじめに

DDR2 SDRAM インターフェイスはソース同期で、読み出しデータと読み出しストローブが同一エッジで送信されます。Virtex-5 FPGA を使用してこの送信データをキャプチャするには、ストローブまたはデータのいずれかを遅延させます。このデザインの場合、読み出しデータは遅延させたストローブドメインでキャプチャされ、ISERDES の FPGA クロック ドメインで再キャプチャされます。ISERDES の OCLK 入力および CLKDIV 入力は共に FPGA の高速クロックから提供されます。つまり、ISERDES の Q3 および Q4 出力は無視されます。BUFIO クロック リソースにアクセスするため、差動ストローブが CC (Clock-Capable) I/O ペアに配置されます。BUFIO クロッキング リソースは、遅延させた読み出し DQS とそれに関連するデータ用の ISERDES のクロック入力をつなぎます。FPGA で送信される書き込みデータおよびストローブは ODDR を使用します。

ここでは、DDR2 SDRAM デバイスの機能について簡潔に説明し、高速 DDR2 メモリにインターフェイスする場合のコントローラ動作について詳細に説明します。また、コントローラへのバックエンド ユーザー インターフェイスの説明も記載しています。

### DDR2 SDRAM の概要

DDR2 SDRAM デバイスは、DDR SDRAM ファミリの次世代デバイスです。この DDR2 SDRAM デバイスでは SSTL 1.8V I/O 規格が使用されます。次のセクションでは、DDR2 SDRAM デバイスが提供する機能を説明し、DDR SDRAM と DDR2 SDRAM の主な相違点について説明します。

DDR2 SDRAM デバイスは、DDR アーキテクチャを活用して高速動作を実現します。メモリは、コントローラが提供する差動クロックを使用して動作します。コマンドは、クロックの立ち上がりエッジごとにラッチされます。双方向のデータ ストローブ (DQS) は、レシーバでのデータ キャプチャ用にデータと一緒に送信されます。DQS は、読み出し中は DDR2 SDRAM デバイスによって、書き込み中はコントローラによって送信されるストローブです。DQS は、読み出しではデータとエッジが揃えられ、書き込みではデータの中央に位置します。

DDR2 SDRAM デバイスへの読み出しおよび書き込みアクセスはバースト転送されます。アクセスは、アクティブ コマンドがレジスタに入ると開始し、それに読み出しまたは書き込みコマンドが続きます。アクティブ コマンドと共にレジスタに入力されたアドレス ビットは、アクセスするバンクおよび行の選択に使用されます。一方、読み出しまたは書き込みコマンドと共にレジスタに入力されたアドレス ビットは、バースト アクセスのバンクおよび列の開始位置の選択に使用されます。

DDR2 コントローラのリファレンス デザインには、書き込みアドレス、書き込みデータ、および読み出しアドレスを作成するユーザー バックエンド インターフェイスが含まれます。この情報は、バックエンド モジュールとコントローラ モジュールのアドレスおよびデータを同期させるため、3 つのバックエンド FIFO に格納されます。アドレス FIFO で利用可能なアドレスに基づき、メモリのタイミング要件を考慮した上で、コントローラはメモリに適切なコマンドを発行します。ロジック ブロックのインプリメンテーションについては、次のセクションで詳細に説明します。

© 2006-2007 Xilinx, Inc. All Rights Reserved. XILINX、Xilinx ロゴ、およびその他本文に含まれる商標名は Xilinx の商標です。本文書に記載されている「Xilinx」、ザイリックスのロゴ、およびザイリックスが所有する製品名等は、米国 Xilinx Inc. の米国における登録商標です。その他に記載されている会社名および製品名等は、各社の商標または登録商標です。保証否認の通知: Xilinx ではデザイン、コード、その他の情報を「現状有姿の状態」で提供しています。この特徴、アプリケーションまたは規格の一実施例としてデザイン、コード、その他の情報を提供しておりますが、Xilinx はこの実施例が権利侵害のクレームを全く受けないということを表明するものではありません。お客様がご自分で実装される場合には、必要な権利の許諾を受ける責任があります。Xilinx は、実装の妥当性に関するいかなる保証を行なうものではありません。この保証否認の対象となる保証には、権利侵害のクレームを受けないことの保証または表明、および市場性に対する適合性についての黙示的な保証も含まれます。

## コントローラが発行する DDR2 SDRAM コマンド

表 1 に、コントローラが発行するコマンドを示します。コマンドは、行アドレス選択 ( $\overline{\text{RAS}}$ )、列アドレス選択、( $\overline{\text{CAS}}$ )、および書き込みイネーブル ( $\overline{\text{WE}}$ ) 制御信号を通してメモリで検知されます。デバイスのコンフィギュレーション後、クロック イネーブル (CKE) は High に維持され、チップ セレクト ( $\overline{\text{CS}}$ ) はデバイスの動作中 Low に維持されます。コントローラがサポートする DDR2 コマンド機能については、[モードレジスタの定義](#)で説明します。

表 1: DDR2 コマンド

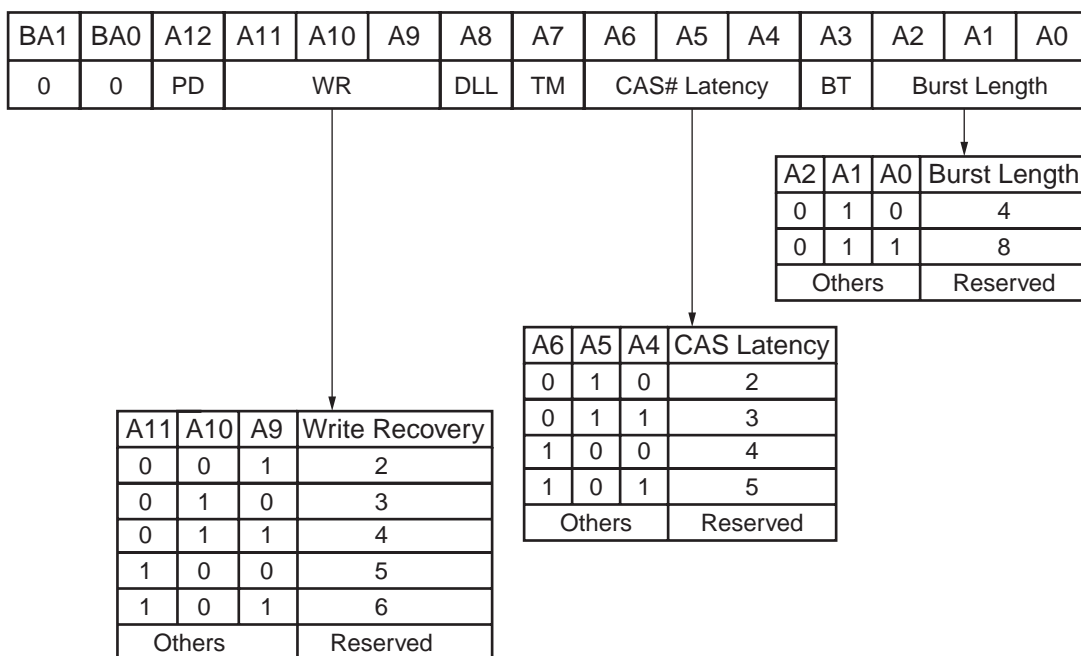
手順	機能	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{WE}}$
1	ロードモード	L	L	L
2	自動リフレッシュ	L	L	H
3	プリチャージ <sup>(1)</sup>	L	H	L
4	バンクの有効化	L	H	H
5	書き込み	H	L	L
6	読み出し	H	L	H
7	動作なし/IDLE	H	H	H

メモ:

1. アドレス信号 A10 は、全バンクのプリチャージ時には High、1つのバンクのプリチャージ時には Low に維持されます。

## モードレジスタの定義

モードレジスタを使用し、DDR2 SDRAM の特定の動作モードを指定します。この指定には、バースト長、バーストタイプ、CAS レイテンシ、および動作モードの選択が含まれます。図 1 に、このコントローラで使用されるモードレジスタの機能を示します。バンクアドレス BA1 および BA0 がモードレジスタを選択します。



X858\_01\_042006

図 1: モードレジスタ

表 2 に、バンク アドレス ビットのコンフィギュレーションを示します。

表 2: バンク アドレス ビットのコンフィギュレーション

BA1	BA0	モード レジスタ
0	0	モード レジスタ (MR)
0	1	EMR1
1	0	EMR2
1	1	EMR3

### 拡張モード レジスタの定義

モード レジスタによって制御される機能以外に、DLL イネーブル/ディスエーブル、出力駆動能力、ODT (オンダイ終端)、ポストッド CAS の AL (追加レイテンシ)、OCD (オフチップドライバのインピーダンス キャリブレーション)、DQS イネーブル/ディスエーブル、RDQS/RDQS イネーブル/ディスエーブル、OUTPUT ディスエーブル/イネーブルが、拡張モード レジスタによって制御されます (表 3)。このリファレンス デザインでは、OCD は使用されません。

表 3: 拡張モード レジスタ

BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	Out	$\overline{\text{RDQS}}$	$\overline{\text{DQS}}$	OCD Program			$R_{\text{TT}}$	Posted CAS			$R_{\text{TT}}$	ODS	DLL

#### 拡張モード レジスタ 2 (EMR2)

バンク アドレス ビットは 10 (BA1 は High、BA0 は Low) に設定されます。アドレス ビットはすべて Low に設定されます。

#### 拡張モード レジスタ 3 (EMR3)

バンク アドレス ビットは 11 (BA1 および BA0 が High) に設定されます。EMR2 と同様に、アドレス ビットはすべて Low に設定されます。

### 初期化シーケンス

コントローラ ステート マシンが使用する初期化シーケンスは、DDR2 SDRAM 仕様に従っています。インターフェイスは、メモリの電圧要件を満たす必要があります。次に、初期化時に発行されるコマンドのシーケンスを示します。

1. 電圧およびクロックの安定後、NOP または Deselect コマンドを 200 $\mu$ s 間適用します。
2. CKE をアサートします。
3. 400ns 後に全バンクのプリチャージ コマンドを実行します。
4. EMR (2) コマンドを実行します。BA0 を Low にし、BA1 を High にします。
5. EMR (3) コマンドを実行します。BA0 および BA1 の両方を High にします。
6. EMR コマンドを実行し、メモリ DLL を使用可能にします。BA1 および A0 を Low にし、BA0 を High にします。
7. モード レジスタの設定コマンドを実行して、DLL をリセットします。DLL をロックするには、クロックの 200 サイクルが必要です。
8. 全バンクのプリチャージ コマンドを実行します。
9. 自動リフレッシュ コマンドを 2 回実行します。
10. モード レジスタの設定コマンドで A8 を Low にして実行し、デバイスの動作を初期化します。

11. EMR コマンドを実行し、ビット E7、E8、E9 を 1 に設定することによって OCD のデフォルト値を有効にします。
12. EMR コマンドを実行し、ビット E7、E8、E9 を 0 に設定することによって、OCD 非選択を有効にします。

初期化シーケンスの完了後、コントローラはダミーの書き込み、それに続いてダミーの読み出しを DDR2 SDRAM メモリに発行します。これにより、データバス モジュールは Virtex-5 の入力遅延ブロック内で適切なタップ数を選択します。データバス モジュールは、必要な遅延タップ数を決定した後、コントローラに対して dp\_dly\_slct\_done 信号をアサートし、コントローラは IDLE ステートに移行します。

#### プリチャージ コマンド

プリチャージ コマンドは、特定のバンクにあるオープンな行を非アクティブにする場合に使用します。このコマンドの発行後、指定時間 ( $t_{RP}$ ) 経過すると、そのバンクの後続行を使用可能にできます。入力 A10 によって、プリチャージするバンクが 1 つか、すべてかを決定します。

#### 自動リフレッシュ コマンド

DDR2 デバイスは  $7.8\mu\text{s}$  ごとにリフレッシュする必要があります。自動リフレッシュ コマンドをフラグする回路は、コントローラ内に作成されています。コントローラは 16 分の 1 のシステム クロックを使用し、リフレッシュ カウンタを駆動します。auto\_ref 信号のアサートによって、自動リフレッシュ コマンドが必要であることが示されます。この auto\_ref 信号は、前の自動リフレッシュ コマンドの  $7.8\mu\text{s}$  後、High になります。その後、コントローラは現在のバーストの完了後、自動リフレッシュ コマンドを発行します。このデザインでは、自動リフレッシュ コマンドが最も優先されます。

#### アクティブ コマンド

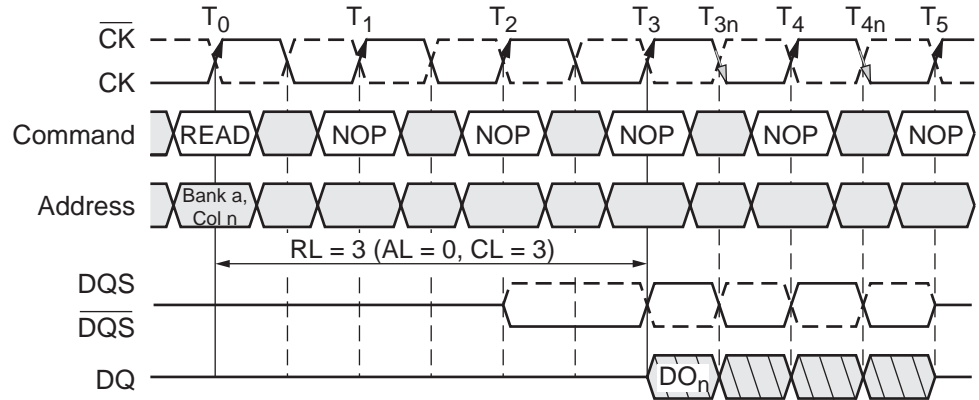
読み出しまたは書き込みコマンドを DDR2 SDRAM メモリのバンクに発行するには、アクティブ コマンドを用いて、該当バンクの行をアクティブにする必要があります。行がオープンになると、 $t_{RCD}$  仕様に従って、読み出しまたは書き込みコマンドが発行可能です。DDR2 SDRAM デバイスは、ポストッド CAS の追加レイテンシもサポートしています。このレイテンシにより、追加レイテンシ クロック サイクルを用いることで内部読み出しまたは書き込みコマンドが実際にレジスタ入力されるタイミングを遅延させて、 $t_{RCD}$  仕様よりも前に発行できるようになります。

コントローラが競合を検出した場合、プリチャージ コマンドを発行し、オープン行を非アクティブにして新しい行に別のアクティブ コマンドを発行します。競合は、入力されているアドレスが現在オープン の行以外の行を参照した場合に発生します。

#### 読み出しコマンド

読み出しコマンドは、アクティブな行へのバースト読み出しアクセスを開始する際に使用されます。BA0 および BA1 の値でバンク アドレスを選択します。A<sub>0</sub> ~ A<sub>i</sub> で与えられるアドレス入力が、開始列の位置を選択します。読み出しバーストの終了後、その行はプリチャージ コマンドが発行されるまで、引き続きアクセスが可能です。

図 2 に、追加レイテンシが 0 の場合の読み出しコマンド例を示します。つまり、この例では読み出しレイテンシが 3 で、CAS レイテンシと同じです。



X858\_02\_042606

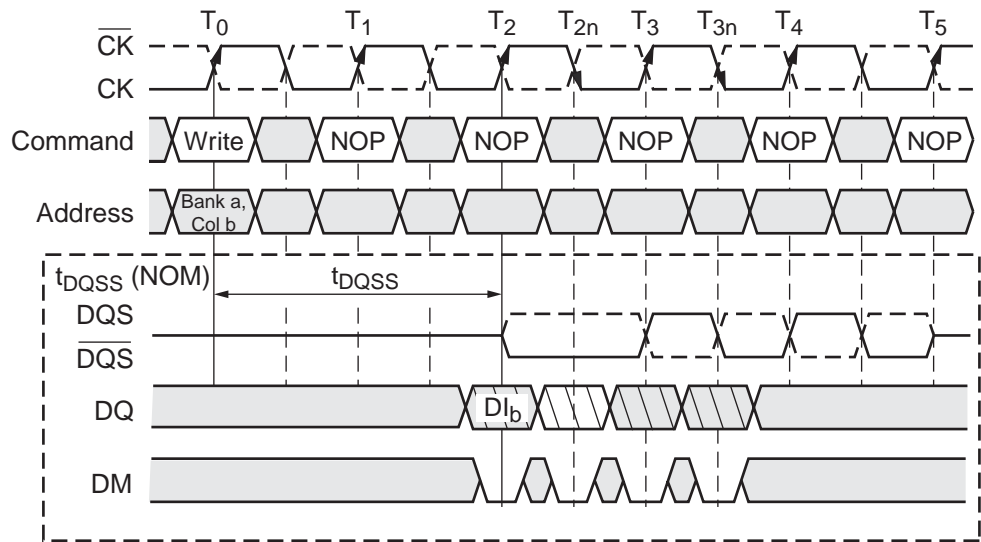
図 2：読み出しコマンドの例

書き込みコマンド

書き込みコマンドは、アクティブな行へのバーストアクセスを開始する際に使用されます。BA0 および BA1 の値がバンクアドレスを選択し、アドレス入力 A<sub>0</sub> ~ A<sub>1</sub> の値が、アクティブな行の開始列の位置を選択します。DDR2 SDRAM では、読み出しレイテンシ (RL) から 1 クロック サイクルを引いた値と等しい書き込みレイテンシ (WL) を使用します。

$$\text{書き込みレイテンシ} = \text{読み出しレイテンシ} - 1 = (\text{追加レイテンシ} + \text{CASレイテンシ}) - 1$$

図 3 に、WL が 2 の場合の書き込みバースト例を示します。書き込みコマンドから DQS 信号の最初の立ち上がりエッジまでの時間は、WL で決定されます。

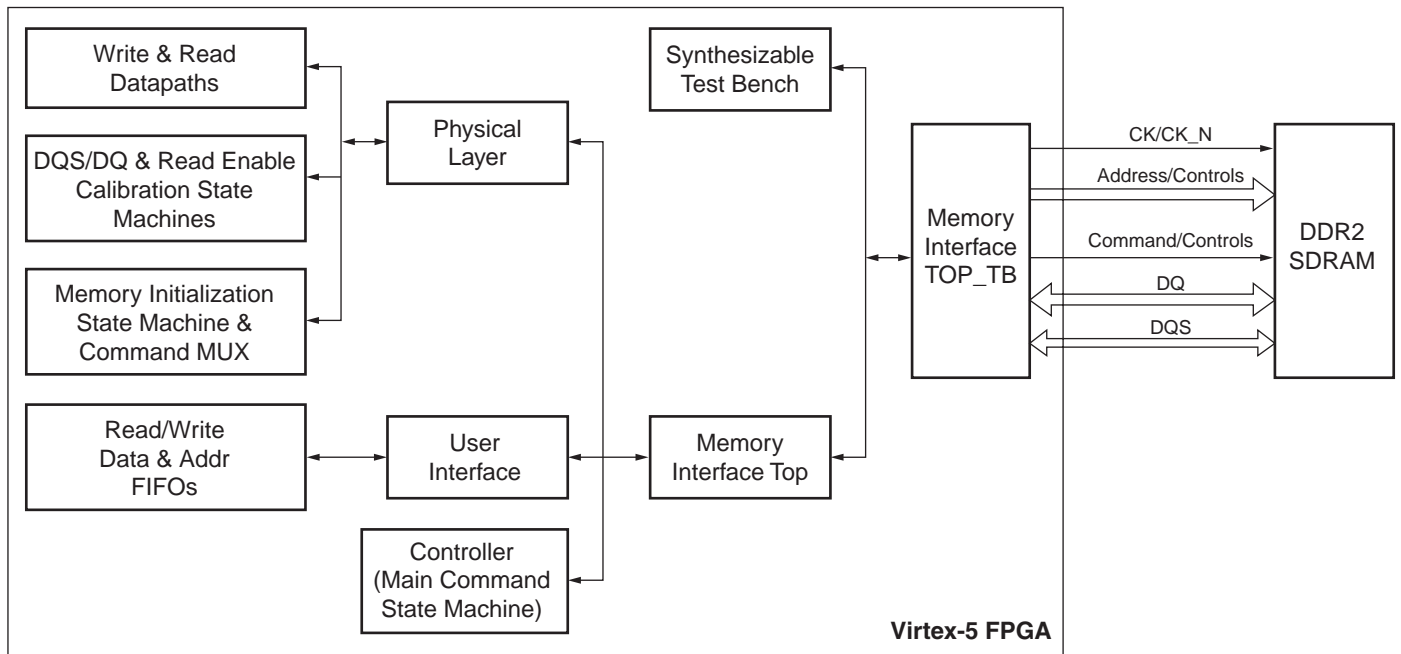


X858\_03\_042006

図 3：書き込みコマンドの例

## DDR2 SDRAM インターフェイス デザイン

図 4 に、DDR2 インターフェイスのブロック図を示します。ユーザー インターフェイスに接続している FIFO すべてが非同期 FIFO のため、バックエンドのユーザー回路は任意の周波数で動作可能です。



X858\_04\_042606

図 4：DDR2 インターフェイス全体のブロック図

### バックエンドの ユーザー回路

バックエンド回路は、メモリ デバイスとメモリ インターフェイス (DDR2 コントローラおよび物理層) 間の読み出しおよび書き込みアクセスをテストするためのアドレス、データ パターンを提供します。バックエンドには、バックエンド ステート マシン、読み出しデータ コンパレータ、およびデータ生成モジュールが含まれます。データ生成モジュールは、メモリに書き込まれる、さまざまなアドレスおよびデータ パターンを作成します。アドレス位置は、ここでは ROM として使用されているブロック RAM に前もって格納されます。格納されたアドレス値が選択され、DDR2 SDRAM デバイスの別の行およびバンクへのアクセスがテストされます。データ パターン生成ブロックには、データ パターンを発行するステート マシンが含まれます。バックエンド ステート マシンは、ユーザー バックエンドをエミュレートします。このステート マシンは、書き込みまたは読み出しイネーブル信号を発行し、データ ジェネレータ モジュールがアクセスする特定の FIFO を決定します。

### ユーザー インター フェイス

バックエンドのユーザー インターフェイスには、アドレス FIFO、書き込みデータ FIFO、読み出しデータ FIFO の 3 つの FIFO があります。最初の 2 つの FIFO はバックエンドのユーザー モジュールによってアクセスされますが、読み出しデータ FIFO は、キャプチャした読み出しデータを保存するため、データパス モジュールによってアクセスされます。

## ユーザーとコントロール間のインターフェイス

表 4 に、ユーザー インターフェイスとコントローラ間の信号を示します。

表 4: ユーザー インターフェイスとコントローラ間の信号

ポート名	ポート幅 (ビット)	説明	メモ
usr_ip_add_fifo_addr	36	ユーザー インターフェイスにあるアドレス FIFO の出力。次のアドレスをマッピング: <ul style="list-style-type: none"> <li>• Memory Address 31:0], (CS, Bank, Row, Column)[</li> <li>• Reserved [33:32]</li> <li>• Command Request [35:34]</li> </ul>	アドレス FIFO へアドレス書き込む際の FULL ステータス フラグをモニタ。
usr_ip_add_fifo_empty	1	ユーザー インターフェイスにあるアドレス FIFO の EMPTY ステータス フラグ出力。この信号がアサートされると、コントローラは FIFO 出力のアドレスを処理する。	FIFO16 EMPTY フラグ。
ctrl_af_rden	1	ユーザー インターフェイスにあるアドレス FIFO への読み出しイネーブル入力。	コントローラのステートが読み出しまたは書き込みの場合、クロックの 1 サイクル間アサートされる。
ctrl_wdf_rden	1	ユーザー インターフェイスにある書き込みデータ FIFO への読み出しイネーブル入力。	書き込みステート後、コントローラによってクロックの 2 サイクル間アサートされる。バースト長が 8 の場合、クロックの 4 サイクル間アサートされる。書き込みコマンドの発行前に、必要なバースト長の書き込みアドレスに関連した書き込みデータ FIFO に十分なデータが必要。たとえば、64 ビットデータバスで、バースト長が 4 の場合、ユーザーは読み出しコマンドの発効前に、書き込みアドレスごとに 2 つの 128 ビットデータワードを入力しなければならない。

メモリ アドレス (Af\_addr) には、列アドレス、行アドレス、バンク アドレス、およびワード数が多いメモリ インターフェイス用のチップ セレクト幅が含まれます (表 5)。

表 5: Af\_addr メモリ アドレス

アドレス	説明
列アドレス	col_ap_width - 1:0
行アドレス	col_ap_width + row_address - 1:col_ap_width
バンク アドレス	col_ap_width + row_address + bank_address - 1:col_ap_width + row_address
チップ セレクト	col_ap_width + row_address + bank_address + chip_address - 1:col_ap_width + row_address + bank_address

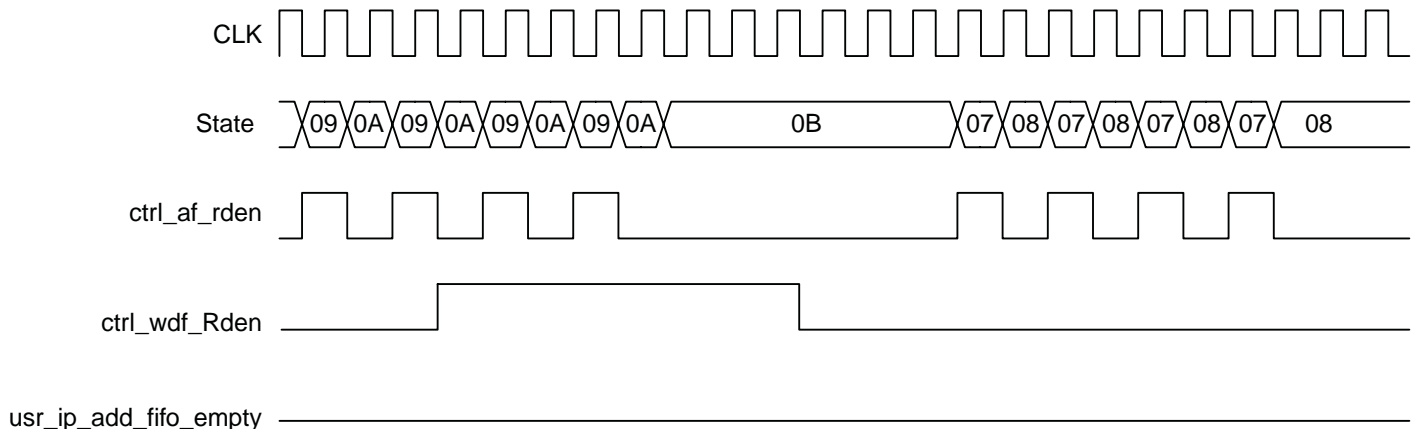
## コマンド要求

表 6 に、読み出しおよび書き込みコマンド要求のフォーマットを示します。

表 6: オプション コマンド

コマンド	説明
00	書き込み
01	読み出し
10	NOP
11	NOP

図 5 に、バースト長が 4 のとき、4 連続の書き込みに続いて 4 連続の読み出しが実行される場合を示します。また、表 7 には図 5 のステート信号の値を示します。



X858\_05\_042606

図 5: バースト長が 4 の場合の連続書き込みと読み出し

表 7: 図 5 のステート信号値

ステート	説明
09	Burst Write
0A	Write Wait
07	Burst Read
0B	Write Read
08	Read Wait

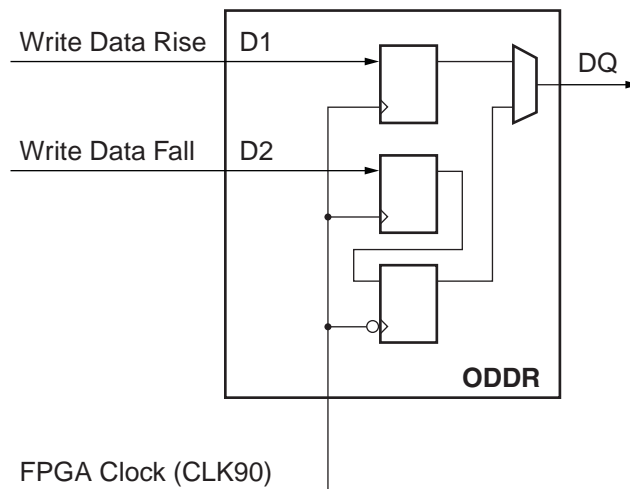


## 物理層

物理層には、書き込みデータパス、読み出しデータパス、DQS および DQ キャリブレーション用のキャリブレーション ステート マシン、読み出しイネーブル調整用のキャリブレーション ロジック、メモリ初期化ステート マシンが含まれます。書き込みデータパスは、書き込みコマンド中に送信されるデータおよびストロブ信号を生成します。そして、読み出しデータパスが、読み出しストロブドメインで読み出しデータをキャプチャします。

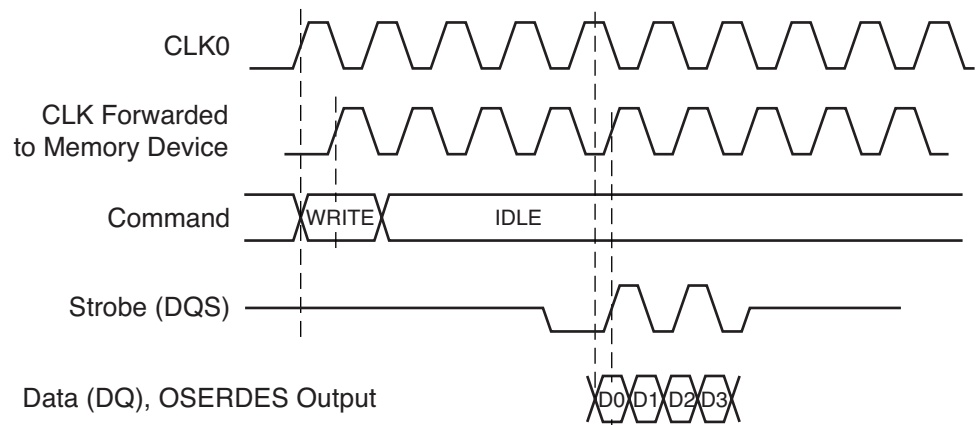
## 書き込みデータパス

書き込みデータパスは、Virtex-5 の全 I/O に内蔵の ODDR を使用します。ODDR はデータ (DQ) およびストロブ (DQS) 信号を送信します。メモリ仕様に従うと、DQS は DQ に対して中央に位置するよう送信されなければなりません。メモリにフォワードされたストロブ (DQS) は位相が CLK0 と 180°ずれています。つまり、図 6 に示すように、ODDR を使用して送信される書き込みデータは CLK90 でクロッキングする必要があります。書き込み DQS および DQ のタイミング図は図 7 に示します。



X858\_06\_042606

図 6： OSERDES を使用した書き込みデータの送信



X858\_07\_041806

図 7： 書き込みレイテンシが 4 の場合のストロブ (DQS) およびデータ (DQ) のタイミング

## 書き込みのタイミング解析

表 8 に、333MHz (667Mb/s) のインターフェイスにおける書き込みタイミング解析を示します。

表 8 : 333MHz での読み出しのタイミング解析

不確定なパラメータ	値	DQS 前の 不確定時間	DQS 後の 不確定時間	説明
$T_{\text{CLOCK}}$	3000			クロック周期。
$T_{\text{MEMORY\_DLL\_DUTY\_CYCLE\_DIST}}$	150	150	150	メモリ DLL からのデューティ サイクルのずれが、クロック位相(クロック周期の半分)から引かれて、 $T_{\text{DATA\_PERIOD}}$ が求められる。
$T_{\text{DATA\_PERIOD}}$	1350			データ周期はクロック周期の半分で、10% のデューティ サイクルのずれはクロック周期から減算されている。
$T_{\text{SETUP}}$	100	100	0	メモリベンダーが指定。
$T_{\text{HOLD}}$	175	0	175	メモリベンダーが指定。
$T_{\text{PACKAGE\_SKEW}}$	30	30	30	DQS の PCB トレース遅延および、それに関連する DQ ビットはパッケージスキューを考慮して調整される。記載値は、誘電率の変動を表す。
$T_{\text{JITTER}}$				DQS および DQ の生成に使用する DCM (デジタルクロックマネージャ) と同じ。
$T_{\text{CLOCK\_SKEW-MAX}}$				グローバルクロックツリースキュー。
$T_{\text{CLOCK\_OUT\_PHASE}}$				同一 DCM の異なる出力間の位相オフセットエラー。
$T_{\text{PCB\_LAYOUT\_SKEW}}$	50	50	50	ボード上のデータラインと関連ストロブ間のスキュー。
総不確定時間				
有効ウィンドウの開始点および終点				
最終的なウィンドウ				

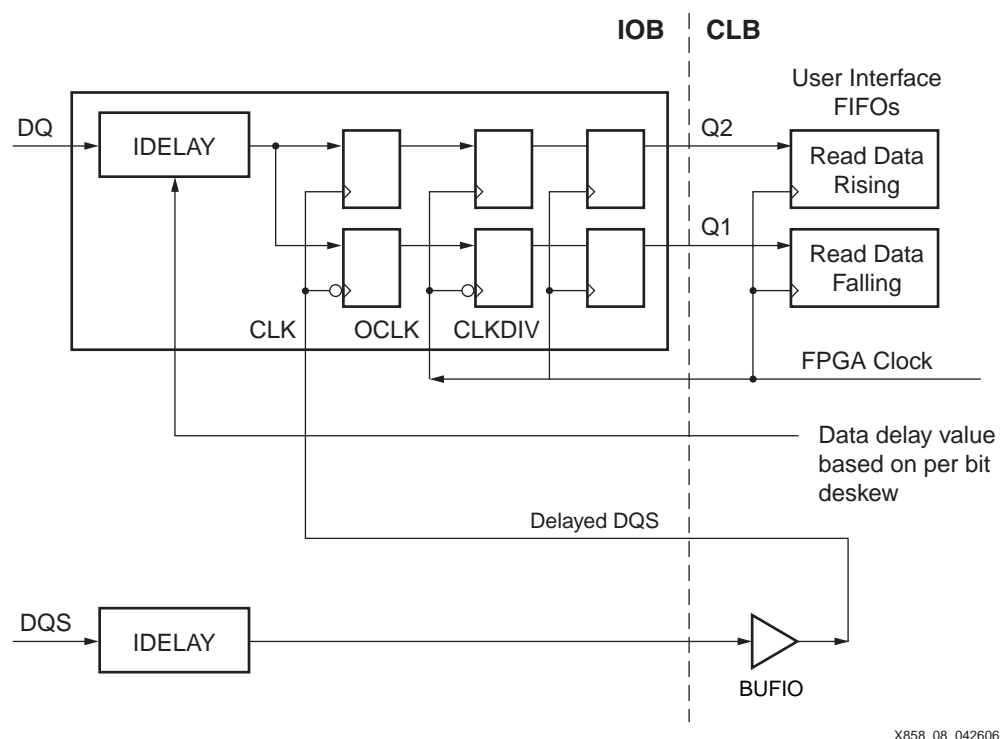
メモ:

1. 同一バンク内の出力フリップフロップと出力バッファ間のスキューは、電圧および温度に対して最小であるとします。

## 読み出しデータパス

読み出しデータパスは、読み出しデータのキャプチャおよび再キャプチャの2段階で構成されています。両段階とも、Virtex-5の全I/Oが備えるISERDESでインプリメントされます。ISERDESにはCLK、OCLK、CLKDIVの3つのクロック入力があります。読み出しデータはCLK (DQS) ドメインでキャプチャされ、OCLK (FPGAの高速クロック) ドメインで再キャプチャされます。そして、最終的にCLKDIV (これもFPGAの高速クロックを使用している) ドメインへ送信され、パラレルデータを提供します。

- CLK: BUFIOを使用して配線された読み出しDQSは、ISERDESのCLKを供給します(図8)。
- OCLK: ISERDESのOCLK入力は、ハードウェアでODDRのCLK入力に接続されます。このデザインでは、CLKfast\_90クロックがISERDES OCLK入力およびODDR CLK入力に供給されます。OCLKに使用されたクロック位相は、書き込みデータに必要な位相で決定されます。
- CLKDIV: 適切に機能するには、OCLKおよびCLKDIVクロック入力の位相が揃えられていることが必須となります。このデザインでは、OCLKおよびCLKDIV入力の両方が同一クロックCLKfast\_90で供給されています。



X858\_08\_042606

図8: IDDRおよびCLBフリップフロップを使用する場合の読み出しデータキャプチャ

### 読み出しのタイミング解析

読み出しデータをエラーなしでキャプチャするには、FPGAクロックドメインにあるフリップフロップのセットアップおよびホールドタイムを満たすよう、読み出しデータおよびストローブを遅延させる必要があります。読み出しデータ(DQ)およびストローブ(DQS)はエッジが揃えられてFPGAで受信されます。BUFIOリソースにアクセスするため、差動DQSペアはCC(Clock-Capable)I/Oペアに配置する必要があります。そして、受信された読み出しDQSはBUFIOリソースを通して、関連したデータビットのISERDESのCLK入力に接続されます。BUFIOおよびクロック配線リソースを通る遅延によって、DQSはデータに対して右にシフトします。

表 9 に、333MHz での読み出しのタイミングを示します。DQS をデータ有効ウィンドウの中央に位置させるために DQ に必要な遅延を決定するには、これらの値が必要です。

表 9: 333MHz での読み出しのタイミング解析

パラメータ	値 (ps)	説明
T <sub>CLOCK</sub>	3000	クロック周期。
T <sub>PHASE</sub>	1500	DDR データのクロック位相。
T <sub>SAMP_BUFIO</sub>	350	Virtex-5-3 スピードグレードデバイスのサンプルウィンドウ。IOB FF のセットアップ/ホールドタイム、クロックジッタ、タップ不確定時間としての 150ps が含まれる。
T <sub>DQD_BUFIO</sub>		BUFIO クロックリソースのデューティサイクルのずれ。
T <sub>DQSQ + T<sub>QHS</sub></sub>	580	メモリの不確定時間のワーストケースで、VT 変動および DQS とそれに関連した DQ 間のスキューを含む。
T <sub>IDELAYTAP_JIT</sub>		20 タップを使用する場合の IDELAY タップの総ジッタ。
総不確定時間		-
ウィンドウ		ワーストケースウィンドウ

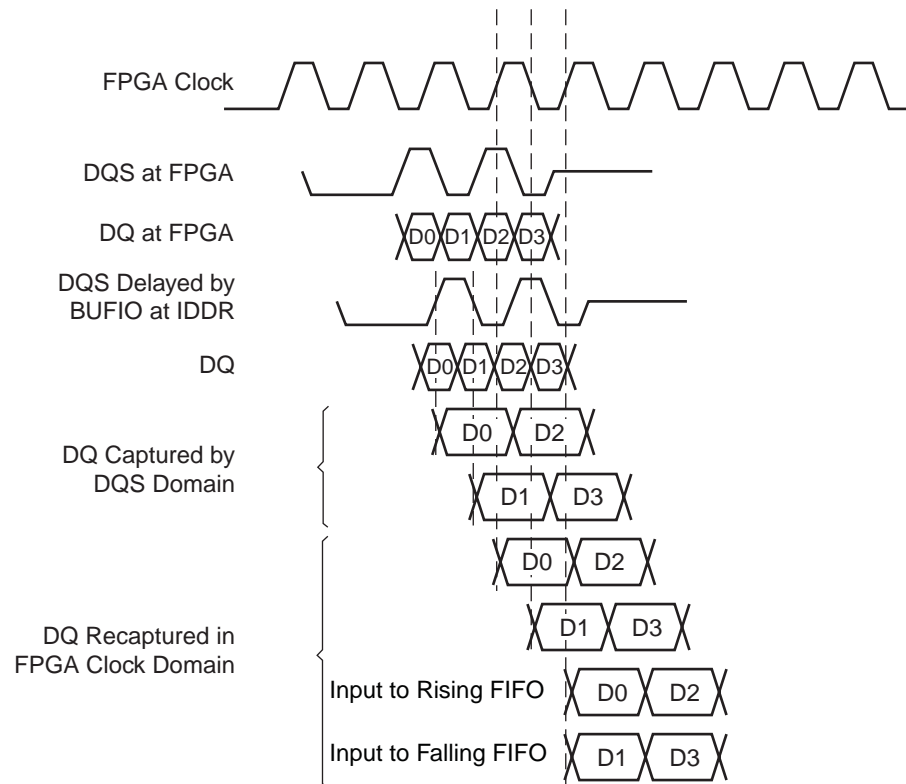
メモ:

1. T<sub>SAMP\_BUFIO</sub> は、BUFIO クロッキングリソースおよび IDELAY を使用する場合、IOB にある DDR 入力レジスタの VT 範囲でのサンプリングエラーです。
2. 表に記載の全パラメータは、ビット単位のキャリブレーション方法を使用する場合、考慮すべき不確定要素です。
3. BUFIO スキュー、package\_skew、pcb\_layout\_skew、および TDQSQ の一部、TQHS はビット単位のキャリブレーション方法では 0 にキャリブレーションされます。内部シンボルの干渉およびクロストークは、動作中のスキューに影響するものですが、この解析では考慮されていません。

## ビット単位のデータスキュー データキャプチャ方法

ISERDES の FPGA クロックドメインで確実なデータキャプチャを行うには、メモリ初期化後にトレーニングシーケンスが必要です。コントローラは、特定のメモリ位置に既知のデータパターンを書き込むため、書き込みコマンドを発行します。コントローラは次に、読み出しコマンドを発行し、この指定された位置から書き込まれたデータをリードバックします。そして、DQ ISERDES 出力の Q1 および Q2 が既知のデータパターンと比較されます。不一致の場合、DQS は 1 タップ分遅延されて再び比較されます。タップは、これが一致するまで 1 つずつインクリメントされます。データ有効ウィンドウが 10 タップ未満の場合、DQ はデータ有効ウィンドウと同タップ数ずつインクリメントされます。これによって、DQ が FPGA のクロックエッジと揃い、FPGA の次のクロックエッジでデータが再キャプチャできます。DQS はタップ単位で遅延され、比較されて、一致するまでタップは増加します。最初に一致が検出された時点で DQS ウィンドウカウントを 1 にし、次に不一致が検出されるまで DQS をタップ単位で遅延させて、これと共に DQS ウィンドウカウントも増加させます。これによって、FPGA クロックドメインのデータ有効ウィンドウ幅を記録します。そして、DQS はウィンドウカウントの半分だけデクリメントされて、DQS のエッジがデータ有効ウィンドウの中央に来るようにします。DQS の位置が固定されると、各 DQ ビットが DQS に対して中央に位置するよう調整されます。DQS および DQ キャリブレーションは、その DQS と関連したすべての DQ ビットのセンタリングが完了すると、終了です。

図 9 に、読み出しデータがストローブドメインでキャプチャされ、ISERDES の FPGA クロックドメインで再キャプチャされる場合のタイミング波形を示します。



X858\_09\_042606

図 9：読み出しデータのキャプチャ タイミング

### コントローラから読み出しデータバスのインターフェイス

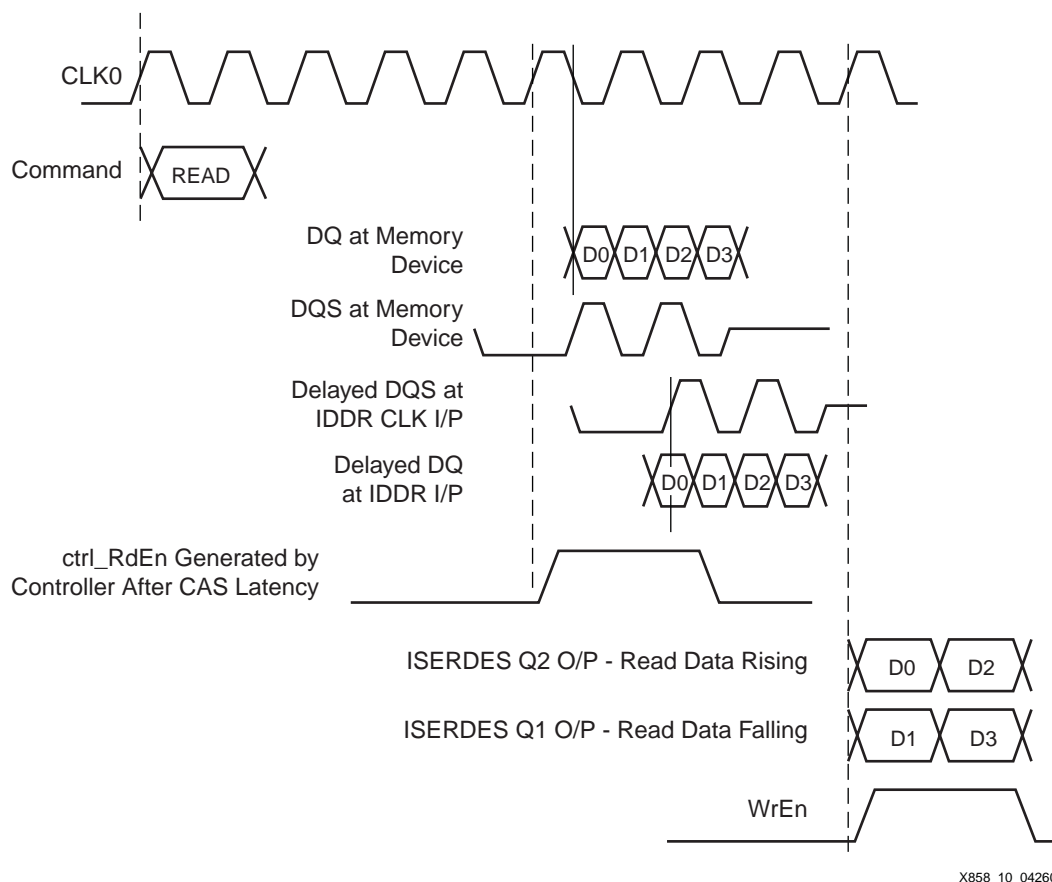
表 10 に、コントローラと読み出しデータバス間の制御信号を示します。

表 10：コントローラと読み出しデータバス間の信号

信号名	信号波 (ビット)	信号の説明	メモ
phy_init_stg1_calib	1	初期化コントローラから読み出しデータバスへの出力。この信号がアサートされると、最初のキャリブレーション (ストローブおよびデータ) が完了。	アサートされている場合、データバスの読み出しデータが有効。
phy_init_stg2_calib	1	初期化コントローラから読み出しデータバスへの出力。この信号がアサートされると、2 番目のキャリブレーション (読み出しイネーブル) が完了。	アサートされている場合、データバスの読み出しデータが有効。
phy_calib_first_calib_done	1	読み出しデータバスから初期化コントローラへの出力で、ストローブおよびデータキャリブレーションが完了したことを示す。	データ、ストローブ、読み出しイネーブルがキャリブレートされると、アサートされる。

表 10: コントローラと読み出しデータバス間の信号 (続き)

信号名	信号波 (ビット)	信号の説明	メモ
phy_calib_second_calib_done	1	読み出しデータバスから初期化コントローラからの出力で、読み出しイネーブルキャリブレーションが完了したことを示す。	読み出しイネーブルのキャリブレーションが完了すると、アサートされる。この信号のアサート後、通常動作が開始する。
ctrl_rden	1	コントローラから読み出しデータバスへの出力で、遅延が標準化された読み出しイネーブル信号用。この信号は、読み出しデータキャプチャ FIFO の書き込みイネーブルとして使用される。	この信号の波形を <a href="#">図 10</a> に示す。これは、バースト長が 4 の場合に CAS レイテンシが 5 で、追加レイテンシが 0 の場合。



X858\_10\_042606

図 10: CAS レイテンシが 5 で、バースト長が 4 の場合の読み出しイネーブル タイミング

DDR2 SDRAM デバイスは、データ有効信号または読み出しイネーブル信号をデータと一緒に出力しないため、読み出しデータを有効にするためには ctrl\_RdEn 信号が必要です。コントローラは、CAS レイテンシおよびバースト長に基づき、この読み出しイネーブル信号を生成します。この読み出しイネーブル信号は CAS レイテンシ後にアサートされ、一連のパイプライン レジスタへの入力となります。読み出しイネーブル信号を ISERDES 読み出しデータ出力と揃えるために要するレジスタの段数は、キャリブレーション中に決定されます。各データ バイトでは、読み出しイネーブル信号が 1 つ生成されます。[図 11](#) に、読み出しイネーブル ロジックのブロック図を示します。

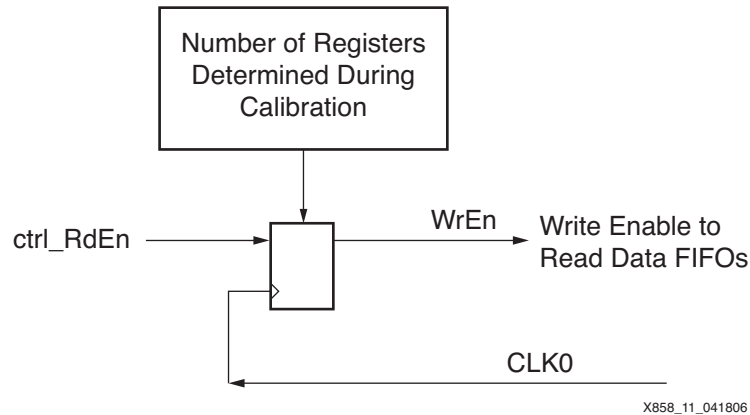


図 11：読み出しイネーブルロジック

## コントローラの インプリメンテーション

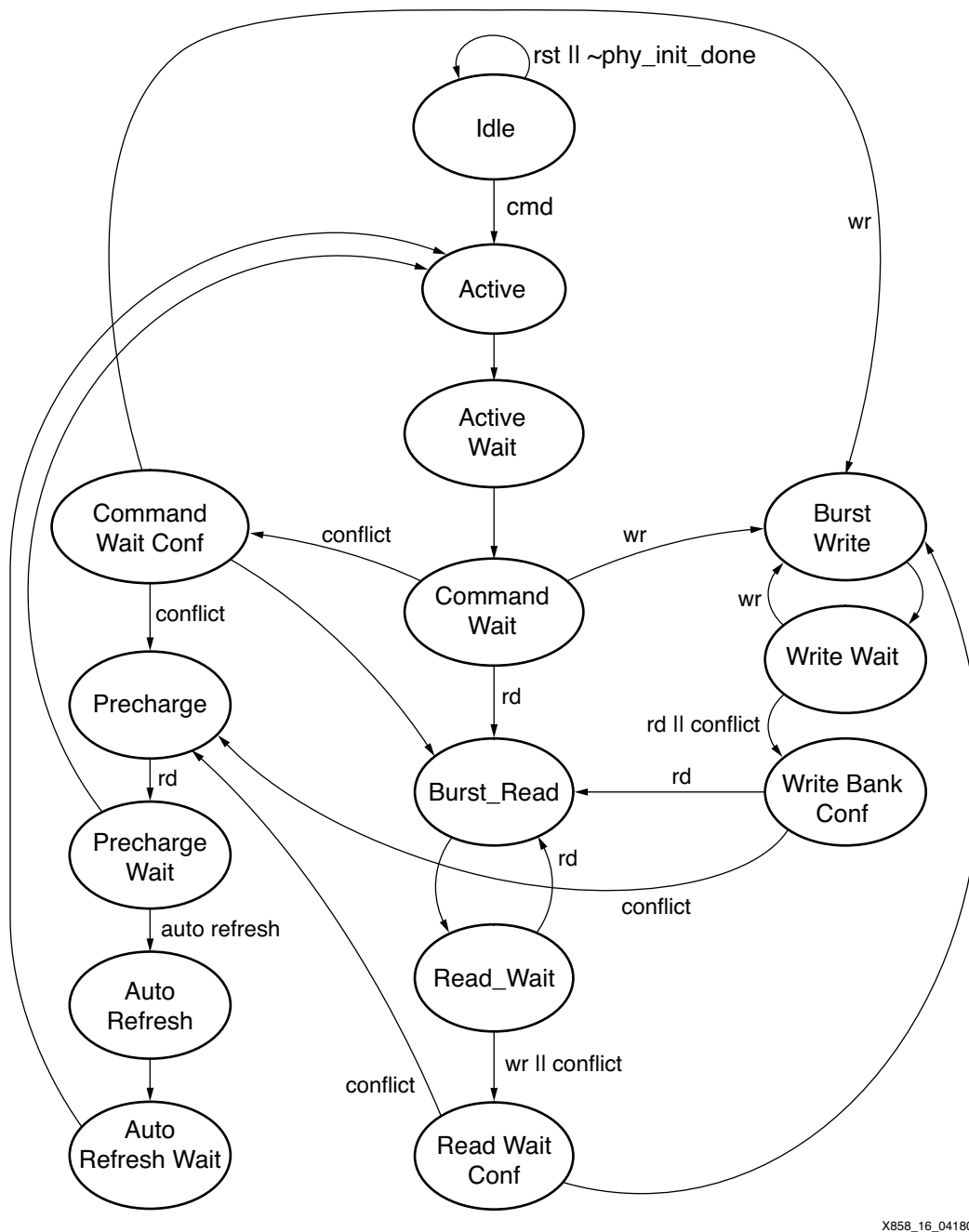
コントローラには、一度に4つのバンクをオープンしておく機能があります。バンクは、コントローラに与えられたコマンド順にオープンになります。4つのバンクがすでにオープンになるときに5つ目のバンクがアクセスされると、最初に使用されたバンクがクローズし、新しいバンクがオープンとなります。自動リフレッシュですべてのバンクがクローズし、コントローラにコマンドが与えられるとオープンになります。

コントローラ ステート マシンは正しいシーケンス順でのコマンドの発行を管理する一方で、メモリのタイミング要件を決定します。

図 12 および次のセクションでは、コントローラ ステート マシンのさまざまなステートについて説明します。

メモリへのコマンド発行前、コントローラは次のような状態にあります。

1. コントローラが FIFO 内のアドレスをデコードします。  
メモ：アドレス FIFO は FWFT (First-Word-Fall-Through) モードです。このモードでは、FIFO に書き込まれた最初のアドレスが FIFO の出力に現れます。
2. 該当バンクの行がオープンでない場合、コントローラはこれをオープンにします。オープンなバンクの別の行にアクセスがあった場合、コントローラはこの行をクローズにし、新たな行をオープンにします。バンクをオープンにした後、コントローラは読み出し/書き込みステートに移行します。
3. コントローラが書き込みステートになった後に読み出しコマンドを受け取ると、読み出しコマンドの発行前に write\_to\_read 時間待機します。同様に、読み出しステートのときにコマンド ロジック ブロックから書き込みコマンドを与えられると、書き込みコマンドの発行前に read\_to\_write 時間待機します。また、読み出しまたは書き込みステートで、コントローラは次のアドレスを取得するためにアドレス FIFO への読み出しイネーブルをアサートします。
4. コマンドは DDR2 メモリへの発行前に、アドレス信号と同期するためパイプライン化されます。



X858\_16\_041806

図 12: DDR2 コントローラ ステート マシン

## リファレンス デザイン

Virtex-5 DDR2 SDRAM メモリ コントローラのリファレンス デザインは、MIG (Memory Interface Generator) ツールに含まれています。また、このツールはザイリンクス CORE Generator™ ソフトウェアに統合されています。最新バージョンのデザインは、次の URL のザイリンクス ウェブサイトから IP アップデートをダウンロードしてください。

[http://japan.xilinx.com/xlnx/xil\\_sw\\_updates\\_home.jsp](http://japan.xilinx.com/xlnx/xil_sw_updates_home.jsp)



## リファレンス デザインの使用リソース数

表 11 に、64-ビット インターフェイスでのリソース使用数を示します。これには、物理層、コントローラ、ユーザー インターフェイス、および合成可能なテストベンチが含まれます。

表 11: 64 ビット インターフェイスのリソース使用数

リソース	使用数	メモ
スライス	2118	コントローラ、合成可能なテストベンチ、ユーザー インターフェイスを含む。
BUFG	4	IDELAY ブロックの 200MHz リファレンス クロック用の BUFG を含む。
BUFIO	8	インターフェイスでのストロープと同数。
DCM	1	-
メモリ デバイス	1	XC5VLX50 が MT9HTF6472Y-667B3 とインターフェイス。

## まとめ

このアプリケーション ノートでは、DDR2 SDRAM コントローラおよび SERDES を使用するデータキャプチャ方法によって、高性能なメモリ インターフェイスに大きなマージンが与えられることを説明しました。この高いマージンは、データを DQS ドメインでキャプチャし、ISERDES で FPGA クロックドメインへ転送することで実現されます。

## 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
2006/05/12	1.0	初版リリース
2007/01/09	1.1	リファレンス デザインへのリンクを更新