



XAPP702 (v1.8) 2007 年 4 月 23 日

Virtex-4 デバイスを使用した DDR2 コントローラ

本資料は英語版 (v1.8) 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

DDR2 SDRAM は DDR SDRAM をしのぐ新機能を備え、400Mb/s 以上のデータ レートを実現したデバイスです。データ レートの高速化に伴い、FPGA のコントローラおよび I/O にもより高い性能が求められます。メモリの動作速度に対応したコントローラを使用することにより、広帯域幅 を達成できます。

はじめに

このアプリケーション ノートでは、Micron DDR2 SDRAM デバイスのインターフェイスとして使用する DDR2 コントローラを Virtex™-4 デバイスでインプリメントする方法を説明します。まず、DDR2 SDRAM デバイス機能の概要を簡単に説明し、次に高速 DDR2 メモリへのインターフェイスとしてのコントローラの使用方法を詳細に説明します。また、コントローラへのバックエンド ユーザー インターフェイスについても解説します。DDR2 SDRAM コントローラのリファレンス デザインは、ザイリンクス CORE Generator に統合されている MIG (Memory Interface Generator) ツールから取得できます。

このアプリケーション ノートでは、ダイレクト クロッキング物理レイヤの DDR2 コントローラの操作を説明しています。Virtex-4 ファミリの物理レイヤ デザインには、ダイレクト クロッキングと SERDES の 2 種類があります。ダイレクト クロッキングの物理レイヤは FPGA クロックを使用して直接データをキャプチャし (データ ストローブを使用せず)、その詳細は、ザイリンクス アプリケーション ノート、[XAPP701](#) 『DDR2 SDRAM Physical Layer Using Direct-Clocking Technique』に記載されています。SERDES 物理レイヤはデータ ストローブを使用してデータをキャプチャし、その詳細は [XAPP721](#) 『High Performance DDR2 SDRAM Interface Data Capture Using ISERDES and OSERDES』に記載されています。

DDR2 SDRAM デバイス概要

DDR2 SDRAM デバイスは SSTL 1.8V I/O 規格を採用し、DDR アーキテクチャを使用することで高速動作を実現しています。メモリは、コントローラが供給する差動クロックを使用して動作します。コマンドは、クロックのすべての立ち上がりエッジで取得されます。双方向データ ストローブ (DQS) は、レシーバでのデータ キャプチャで使用するため、データと共に送信されます。DQS は読み出しでは DDR2 SDRAM デバイスによって、また書き込みではコントローラによって送信されます。読み出しではデータのエッジに揃えられ、書き込みではデータの中央に揃えられます。

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対応です。ACTIVE コマンドが取得されるとアクセスが開始し、その後、READ コマンドまたは WRITE コマンドが実行されます。アクセスするバンクおよび行は、ACTIVE コマンドと共に取得されたアドレス ビットを使用して選択され、バースト アクセスの開始列およびバンクは、READ コマンドまたは WRITE コマンドと共に取得されたアドレス ビットを使用して選択されます。

DDR2 コントローラのデザインには、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生成するユーザー バックエンド インターフェイスが含まれています。この情報は 4 つの非同期 FIFO に格納され、バックエンド モジュールとコントローラ モジュール間のアドレスおよびデータ同期に使用されます。コントローラは、FIFO のデータの有無とコマンドロジックブロックが発行するコマンドに応じて、メモリのタイミング要件を考慮しながら、メモリに正しいコマンドを発行します。

© Copyright 2008 Xilinx, Inc. All rights reserved. Xilinx, the Xilinx logo, the Brand Window, Virtex, Spartan, CoolRunner, ISE, and other designated brands included herein are trademarks of Xilinx, Inc. Certain other third-party trademarks are used under license, for further information, see <http://japan.xilinx.com/legal.htm>. All other trademarks are the property of their respective owners. (この日本語訳 (参考のみ) は、<http://japan.xilinx.com/support/documentation/disclaimer.htm> を参照してください。)

コントローラの DDR2 コマンド

表 1 に、コントローラにより発行されるコマンドを示します。各コマンドは、次の制御信号を使用したメモリで検知されます。

- 行アドレス セレクト ($\overline{\text{RAS}}$)
- 列アドレス セレクト ($\overline{\text{CAS}}$)
- 書き込みイネーブル ($\overline{\text{WE}}$) 信号
- クロック イネーブル(CKE) (デバイスのコンフィギュレーション中および後は High に保持)
- チップ セレクト ($\overline{\text{CS}}$) (デバイスの動作中は Low に保持)

表 1: DDR2 コマンド

選択肢	機能	行アドレスの セレクト信号	列アドレスの セレクト信号	書き込み イネーブル信号
1	Load Mode	L	L	L
2	Auto-Refresh	L	L	H
3	Precharge ⁽¹⁾	L	H	L
4	Bank Activate	L	H	H
5	Write	H	L	L
6	Read	H	L	H
7	No Operation/IDLE	H	H	H

メモ:

1. アドレス信号 A10 は、すべてのバンクをプリチャージするときは High に保持し、1 つのバンクをプリチャージするときは Low に保持します。

モードレジスタ

モードレジスタ (MR) は、DDR2 SDRAM の動作モードを指定します。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を指定できます。図 1 に、コントローラが使用するモードレジスタの機能を示します。

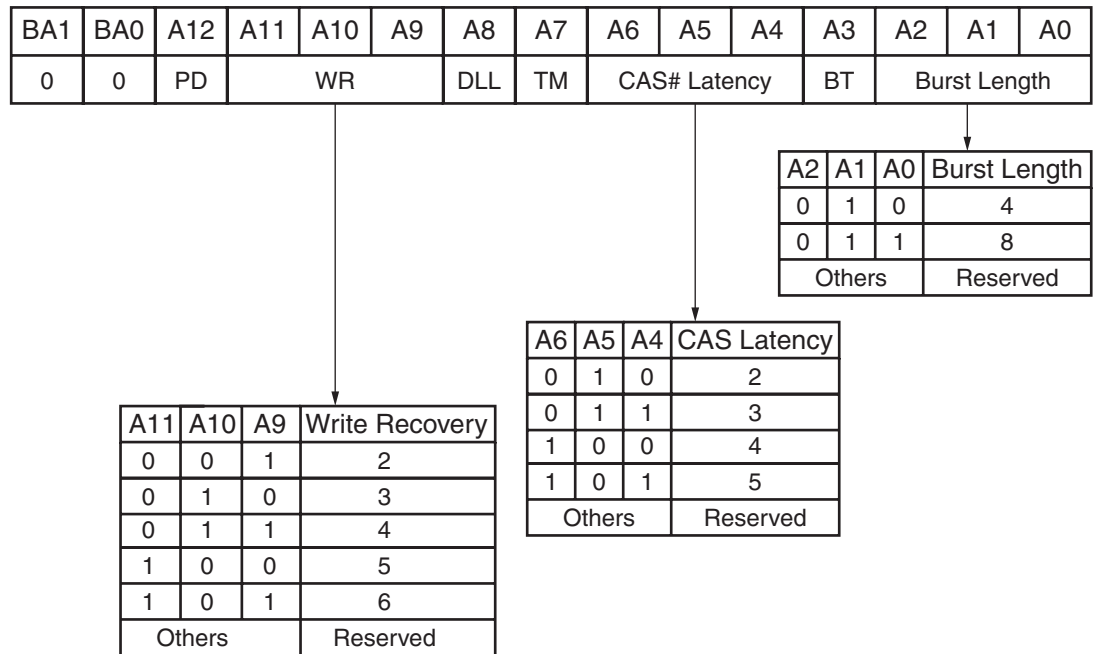


図 1: モードレジスタ

バンクアドレス BA1 および BA0 は、モードレジスタを選択します。表 2 には、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 コンフィギュレーション

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

拡張モードレジスタ

拡張モードレジスタ (EMR) は、MR で制御できない機能を設定できます。設定できる機能は、表 3 に示すとおり、DLL イネーブル/ディスエーブル、出力駆動電流、オンチップ終端 (ODT)、Posted CAS Additive Latency (AL)、オフチップドライバインピーダンス調整 (OCD)、 \overline{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、出力ディスエーブル/イネーブルです。

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	RDQS	\overline{DQS}	OCD Program			R _{TT}	Posted CAS			R _{TT}	ODS	DLL

DDR2 SDRAM リファレンス デザインでは、AL は 0 です。このリファレンス デザインでは OCD は使用しません。

拡張モード レジスタ 2 (EMR2)

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

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 は High) に設定されています。アドレス ビットはすべて Low に設定されています。

初期化シーケンス

コントローラ ステート マシンで使用される 初期化シーケンスは、DDR2 SDRAM 仕様に従っています。インターフェイスには、メモリの電圧の要件を適用してください。電力が安定し、クロック (CK、CK#) が有効になった後、最低 200 μ s 間 NOP または DESELECT コマンドが適用され、クロック イネーブル (CKE) が High にアサートされます。次に初期化時に発行されるコマンドを示します。

1. CKE が High にアサートされた後 400 ns 以上経過すると、PRECHARGE ALL コマンドが発行されます。
2. BA0 および BA2 が Low に、BA1 が High になり、EMR(2) コマンドが発行されます。
3. BA0 および BA1 が High に、BA2 が Low になり、EMR(3) コマンドが発行されます。
4. A0、BA1 および BA2 が Low に、BA0 が High になり、EMR コマンドが発行され、メモリ DLL がイネーブルになります。
5. A8 が High に、BA2、BA1 および BA0 が Low になり、MODE REGISTER SET (MRS) コマンドが発行され、メモリ DLL がリセットされます。
6. PRECHARGE ALL コマンドが発行されます。
7. AUTO-REFRESH コマンドが 2 回発行されます。
8. A8 が Low になり、MRS コマンドが発行されます。このコマンドは DLL をリセットせずにデバイスを操作パラメータで初期化するために必要です。
9. 手順 8 の後 200 クロック 以上経過すると、OCD Calibration (オフチップ ドライバ インピーダンス調整) が実行されます。OCD Calibration を使用しない場合は、デフォルトの OCD コマンド (A9=A8=A7=1) を使用する EMR が発行され、続いて OCD を使用した EMR が発行されます。

これで DDR2 SDRAM デバイスでのキャリブレーションの準備が整いました。

ダイレクト クロッキング物理レイヤで使用されているキャリブレーションはビットごとに行われます。すべての DQ チャネルは IDELAY というプリミティブを使用して FPGA で配線されます。IDELAY は選択可能なタップ遅延ラインから構成されています。このタップ遅延ラインを使用し、ユーザーは DQ ラインに 78 ps の遅延を増分加減することができます。このプロセスを利用し、ユーザーは FPGA クロックに対して中央に揃うよう DQ ラインを最適配置することができます。このキャリブレーションのアルゴリズムおよび回路図は、ザイリンクス アプリケーション ノート XAPP701 に記載されています。

DQ キャリブレーションが完了すると、コントローラにより、既知のパターンをメモリ デバイスに書き込むためのコマンドが発行されます。続いて、書き込みと同じ位置に対して READ コマンドが発行されます。読み出されるデータを基に、READ コマンドとデータ受信の間のクロック サイクルの遅延が決定されます。この数値は格納され、通常の読み出し動作中のリード イネーブルの生成に使用されます。

このリード イネーブル キャリブレーションは data_write および rd_data モジュール内でインプリメントされます。パターンの書き込みでは、コントローラはメモリに WRITE コマンドを発行し、既知の書き込みパターンが data_write モジュール内に生成されます。パターン読み出しでは、READ コマンドが書き込みと同じ位置に発行され、メモリから受信された読み出しデータに揃えて遅延した ctrl_rden 信号が生成されます。この遅延ロジックは pattern_compare8 モジュール内でインプリメントされます。遅

延した `ctrl_rden` は、通常の読み出し中に使用されます。`pattern_compare8` モジュールは `COMP_DONE` 信号をアサートし、リード イネーブル キャリブレーションが完了したことをコントローラに知らせます。これでコントローラ モジュールによる通常の読み出し/書き込み動作の準備が整います。

MIG (Memory Interface Generator) ツール (バージョン 1.7) では、1 つのバンクに対し 1 つの `read_enable` ロジック (`pattern_compare8` モジュール) が生成されます。バンクに 4 データ ビットあれば、`pattern_compare4` モジュールが生成されます。

PRECHARGE コマンド

PRECHARGE コマンドは、特定のバンクの現在アクティブな行を非アクティブにします。コマンドの発行後、指定された時間 (t_{pp}) が経過するまでは、バンク内で別の行をアクティブにできます。入力 A10 は、1 つのバンクをプリチャージするか、すべてのバンクをプリチャージするかを指定します。

AUTO-REFRESH コマンド

DDR2 デバイスは、7.8 μ s おきにリフレッシュする必要があります。コントローラは、自動リフレッシュ カウンタを要求する回路を内蔵しており、DCM の CLKDV 出力を使用して、自動リフレッシュ カウンタに必要な低周波クロックを供給します。DCM の CLKDV 出力 が使用する BUFG を節約する場合は、DCM の 高周波の CLK0 出力 または DCM の CLK/4 出力 (IDELAY 回路で使用) をリフレッシュ カウンタに供給します。自動リフレッシュ回路のクロックを変更する場合は、それに応じて `mem_interface_top_parameters_0.v` ファイルの `max_ref_count` を変更する必要があります。

`auto_ref` 信号は、コントローラに対して、保留中の AUTO-REFRESH コマンドを発行するように要求します。この信号は、AUTO-REFRESH コマンドが発行されるまで High に保持されます。コントローラは、その時点でアクティブなバンクのトランザクションを完了してから、AUTO-REFRESH コマンドを発行します。

ACTIVE コマンド

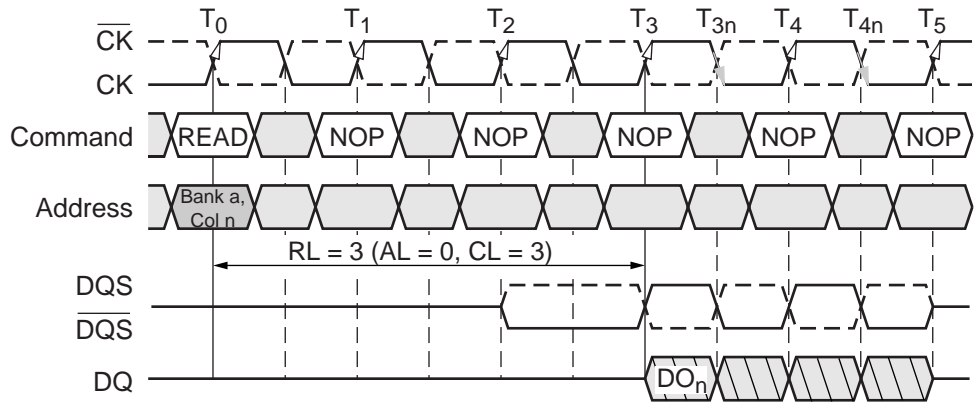
DDR2 SDRAM 内のバンクに READ または WRITE コマンドが発行される前に、ACTIVE コマンドでバンク内の 1 行をアクティブにする必要があります。行がアクティブになると、 t_{RCD} 仕様に従って、その行に対して READ または WRITE コマンドを発行できるようになります。また、DDR2 SDRAM デバイスは Posted CAS AL をサポートしています。実際の READ または WRITE コマンドの内部デバイスへの取得を AL クロック サイクル分遅延させて、 t_{RCD} タイムの前に READ または WRITE コマンドを発行することができます。ACTIVE コマンドに続いて READ または WRITE コマンドが発行されている間、DDR2 コントローラは CAS AL をサポートします。

コントローラは、バンク内の非アクティブな行のアドレスを検知すると、アドレス競合信号を送信し、PRECHARGE コマンドを発行してアクティブな行を非アクティブにし、新しい行に ACTIVE コマンドを発行します。

READ コマンド

READ コマンドは、アクティブな行に対してバースト読み出しアクセスを開始します。BA0 および BA1 の値でバンク アドレスが選択され、アドレス入力 $A_0 \sim A_1$ で開始列を選択します。読み出しバーストの完了後も、プリチャージされるまで、行はアクティブのままです。

図 2 に、READ コマンドで AL が 0 の例を示します。この場合の読み出しレイテンシは、CAS レイテンシ (CL = 3) と同じになります。



x702_02_062804

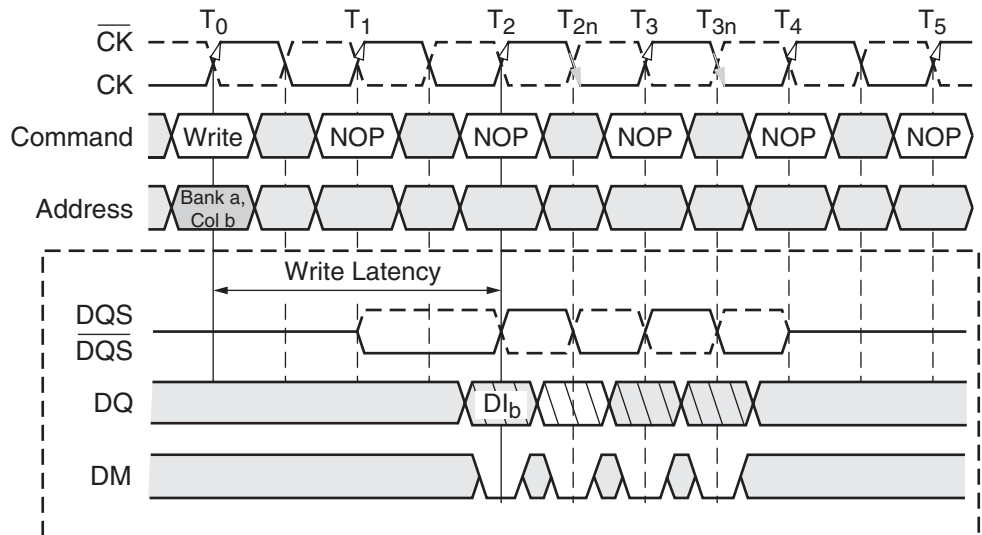
図 2： READ コマンド例

WRITE コマンド

WRITE コマンドは、アクティブな行に対してバースト書き込みアクセスを開始します。BA0 および BA1 はバンク アドレスを指定し、アドレス 入力 A₀ ~ A₁ はアクティブ行の開始列の位置を指定します。DDR2 SDRAM で使用される書き込みレイテンシ (WL) の値は、読み出しレイテンシから 1 クロック サイクルを差し引いた値と同じです。

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

図 3 に、書き込みレイテンシが 2 の書き込みバーストの波形を示します。WRITE コマンドから次の DQS 信号の立ち上がりエッジまでの時間は、書き込みレイテンシによって決まります。

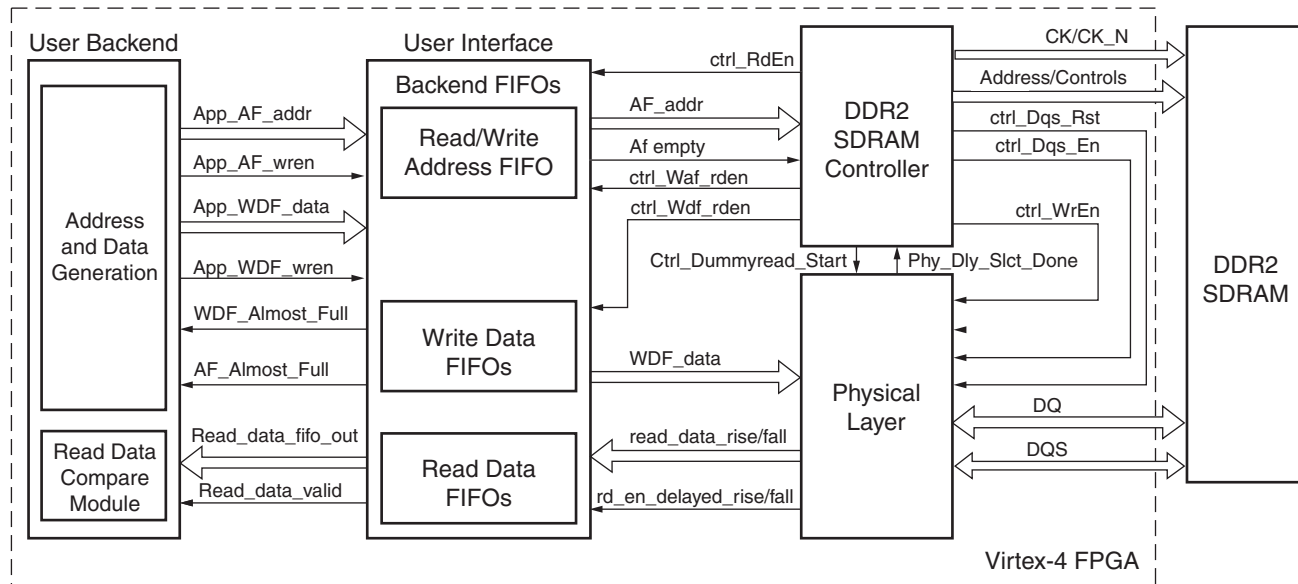


x702_03_022007

図 3： WRITE コマンド例

DDR2 インターフェイスのインプリメント

図 4 に、DDR2 SDRAM インターフェイスのブロック図を示します。



X702_04_021606

図 4：DDR2 SDRAM インターフェイスのブロック図

ユーザー バックエンド

リファレンス デザインのバックエンドから供給されるアドレス パターンおよびデータ パターンを使用して、DDR2 コントローラ デザインをいろいろな角度からテストできます。ユーザー バックエンドには、バックエンド ステート マシン、読み出しデータのコンパレータ、アドレスおよびデータ生成モジュールの各ブロックが含まれます。

アドレスおよびデータ生成モジュールは、メモリに書き込むアドレス パターンやデータ パターンを生成します。アドレス値およびダイナミック コマンド要求は、ROM として使用されるブロック RAM にあらかじめ保存されます。この保存された値を使用して、DDR2 SDRAM デバイスのさまざまな行やバンクへのアクセスをテストします。データ パターン ジェネレータには、立ち上がりエッジデータのパターンを生成するステート マシンが含まれています。立ち下がりエッジデータは、立ち上がりエッジデータの対称形となります。バックエンド ステート マシンは、FIFO をアドレス指定する読み出しイネーブル信号を送ることでユーザー バックエンド アプリケーションをエミュレートします。

ユーザー インターフェイス

バックエンド ユーザー インターフェイスには、書き込みデータ FIFO、読み出しアドレスおよび書き込みアドレス FIFO、読み出しデータ FIFO の 3 つの FIFO が含まれます。書き込みデータ FIFO と読み出しアドレスおよび書き込みアドレス FIFO の 2 つは、ユーザー バックエンド モジュールでアクセスします。読み出しデータ FIFO は、データパス モジュールでアクセスし、取り込んだ読み出しデータを保存します。

ユーザーからコントローラへ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で使用される信号を示します。

表 4: ユーザー インターフェイスとコントローラの間で使用される信号 (ポート)

ポート名	ポート幅	ポートの説明	メモ
Af_addr	36	ユーザー インターフェイスのアドレス FIFO の出力。次のアドレスをマップします。 <ul style="list-style-type: none"> メモリ アドレス (CS、バンク、行、列): [31:0] 予約済み: [35] ダイナミック コマンド要求: [34:32] 	FIFO full ステータス フラグをモニタし、アドレス FIFO にアドレスを書き込みます。
Af_empty	1	ユーザー インターフェイス アドレス FIFO empty ステータス フラグ出力。この信号がアサートされると、書き込みデータの FIFO full ステータス フラグがアサートされるまで、ユーザー アプリケーションでアドレス FIFO に書き込み可能です。	FIFO16 の Almost Empty フラグ
ctrl_af_RdEn	1	ユーザー インターフェイスのアドレス FIFO への読み出しイネーブル入力	この信号は、コントローラのステートが書き込み、読み出し、モードレジスタの読み込み、すべてをプリチャージ、自動リフレッシュ、あるいはダイナミックコマンド要求の結果アクティブとなった場合、1 クロック サイクル間アサートされます。図 5 に、4 回連続する書き込みの後に読み出しが 4 回が続く、バースト長 8 の場合のタイミング波形を示します。
ctrl_wdf_RdEn	1	ユーザー インターフェイスの書き込みデータ FIFO への読み出しイネーブル入力	コントローラは、最初の書き込みステートに遷移後にこの信号を 2 クロック サイクルアサートします。この信号は、バースト長 4 の場合は 2 クロック サイクル間、バースト長 8 の場合は 4 クロック サイクル間アサートされます。図 5 に、タイミング波形を示します。WRITE コマンド送信前に、書き込みデータ FIFO にバースト長分のデータが必要です。たとえば、64 ビット データ バスおよびバースト長 4 の場合、WRITE コマンド送信前に、各書き込みアドレスに対して、2 つの 128 ビット データワードが書き込みデータ FIFO に入力されている必要があります。

表 5 に、メモリ アドレス (Waf_addr) をリストします。これはディープ メモリ インターフェイスの列アドレス、行アドレス、バンク アドレス、チップ セレクトが含まれます。

表 5: Waf_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]

Af_addr のアドレス スペースは連続していません。特に、ユーザー インターフェイスのアドレス バスのビット Af_addr [10] はコントローラ ロジックにより無視されます。メモリ コントローラが 9 列の

ビットのみ DDR2 デバイスにインターフェイスしている場合、Af_addr[9] も無視されます。列アドレス幅のパラメータ col_ap_width には自動プリチャージビット (A10) と列アドレス パラメータが含まれます。列アドレス パラメータは、選択されたメモリ コンポーネントの列アドレス ビット数を示します。

- 9 ビット列アドレスの場合、col_ap_width は 11 と定義されます。下位 9 ビットは列アドレスとなりますが、A9 は使用されません。A10 は通常の読み出しおよび書き込み操作の間 Low に固定されます。このため自動プリチャージ機能はサポートされません。col_ap_width パラメータは PRECHARGE コマンドの間 A10 ビットとして内部で使用されます。
- 10 ビット列アドレスの場合、col_ap_width は 11 として定義されます。
- 11 ビット列アドレスの場合、col_ap_width は 12 として定義されます。

ダイナミック コマンド要求

表 6 に、コントローラの通常動作には不要なコマンドを示します。アプリケーションで必要であれば、使用することもできます。

表 6: コマンドのオプション

コマンド	説明
000	LOAD MODE REGISTER
001	AUTO-REFRESH
010	PRECHARGE ALL
011	ACTIVE
100	WRITE
101	READ

図 5 に、バースト長 8 の、4 連続書き込みとそれに続く 4 連続読み出しを示します。

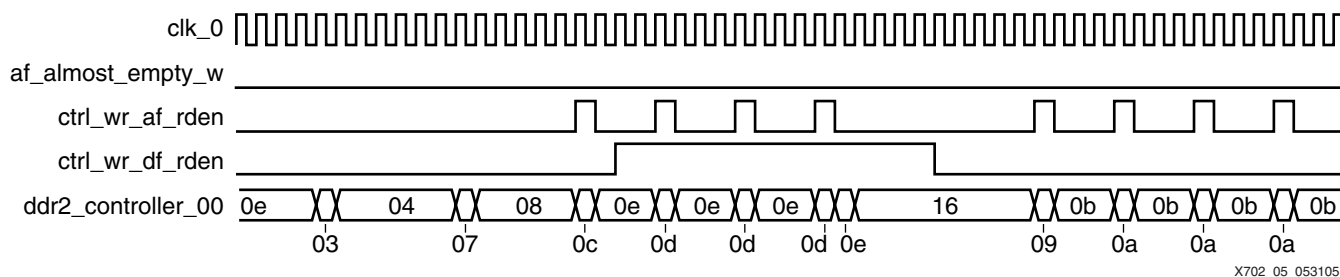


図 5: バースト長 8 の連続書き込みとそれに続く連続読み出し

表 7 に図 5 のステート信号値を示します。

表 7: 図 5 のステート信号値

ステート信号値	説明
0c	最初の書き込み
0d	バースト書き込み
09	最初の読み出し
0a	バースト読み出し
0e	書き込み待ち
03	プリチャージ
04	プリチャージ待ち

表 7: 図 5(続き)のステート信号値

ステート信号値	説明
07	アクティブ
08	アクティブ待ち
16	書き込みから読み出しへ
0b	読み出し待ち

コントローラから物理レイヤへのインターフェイス

表 8 に、コントローラと物理レイヤの間で使用される信号を示します。

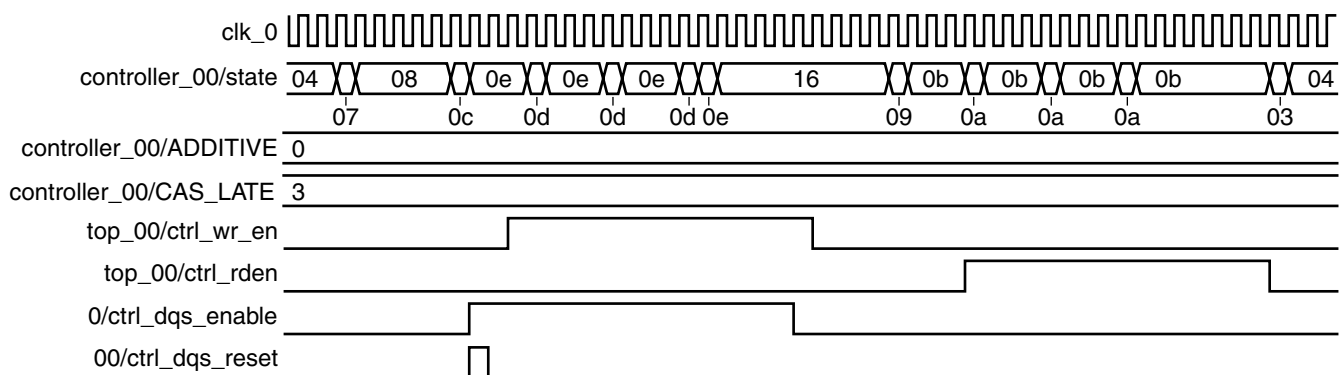
表 8: コントローラと物理レイヤの間で使用される信号 (ポート)

ポート名	ポート幅	ポートの説明	メモ
ctrl_Dummyread_Start	1	コントローラから物理レイヤへの出力。アサートされると、メモリ初期化後に物理レイヤがストロープおよびデータのキャリブレーションを開始します。	ダミーの読み出し状態で読み出しストロープがトグルを開始すると、この信号がアサートされます。phy_Dly_Slct_Done 信号がアサートされると、この信号はディアサートされます。
phy_Dly_Slct_Done	1	キャリブレーションの完了を示す物理レイヤからコントローラへの出力	データビットを遅延して FPGA のグローバルクロックが中央に揃えられると、この信号がアサートされます。phy_Dly_Slct_Done 信号がアサートされると、ctrl_Dummyread_Start 信号はディアサートされず、通常の操作は、この信号のアサート後に開始されます。
ctrl_Dqs_Rst	1	書き込みストロープブリアンプルのコントローラから物理レイヤへの出力	この信号は、書き込み中に 1 クロックサイクル間アサートされず、CAS レイテンシおよび AL 値によって、最初の書き込みステート後、何サイクル後にこの信号がアサートされるかが決定されます。図 6 に、CAS レイテンシが 3、AL 値が 0 で、バースト長 8 で書き込みが 4 回連続した場合のタイミング波形の例を示します。

表 8 : コントローラと物理レイヤの間で使用される信号 (ポート)(続き)

ポート名	ポート幅	ポートの説明	メモ
ctrl_Dqs_En	1	コントローラから物理レイヤへの書き込みストロブの出力	この信号は、バースト長 4 の書き込みでは 3 クロック サイクル、およびバースト長 8 の書き込みでは 5 クロック サイクルの間アサートされます。CAS レイテンシおよび AL 値によって、最初の書き込みまたはバースト書き込みステート後、何サイクル後にこの信号がアサートされるかが決定されます。図 6 に、CAS レイテンシが 3、AL 値が 0 で、バースト長 8 で書き込みが 4 回連続した場合のタイミング波形を示します。
ctrl_WrEn	1	書き込みデータのトライステート コントローラ用のコントローラから物理レイヤへの出力	この信号は、バースト長 4 の書き込みでは 2 クロック サイクル、バースト長 8 の書き込みでは 4 クロック サイクルの間アサートされます。CAS レイテンシおよび AL 値によって、最初の書き込みまたはバースト書き込みステート後、何サイクル後にこの信号がアサートされるかが決定されます。図 6 に、CAS レイテンシが 3、AL 値が 0 で、バースト長 8 で書き込みが 4 回連続した場合のタイミング波形を示します。

図 6 にコントローラから物理レイヤへの制御信号のタイミング波形を示します。



X702_06_061605

図 6 : コントローラから物理レイヤへの制御信号のタイミング波形

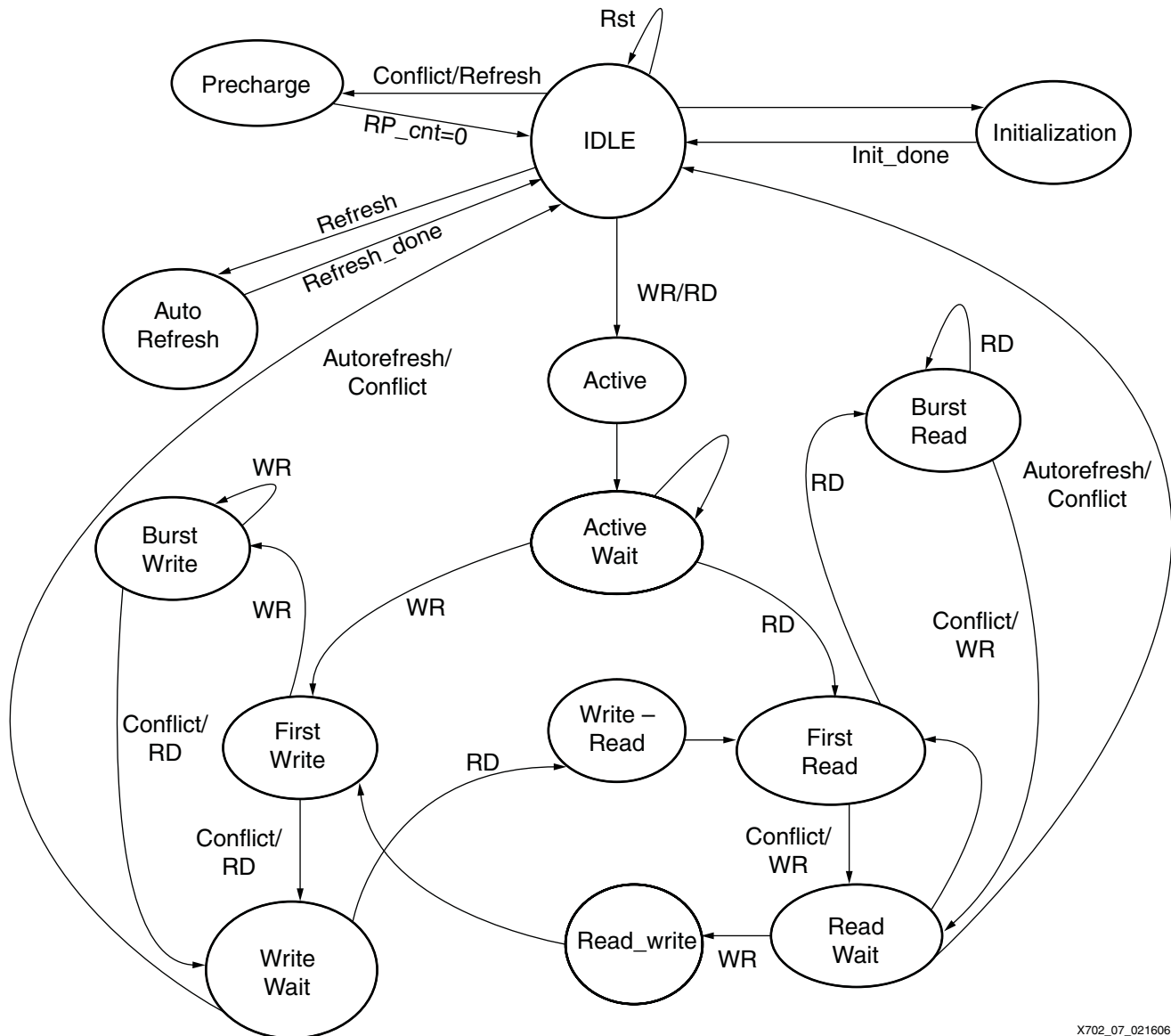
表 9 に図 6 のステート信号値を示します。

表 9: 図 6 のステート信号値

ステート信号値	説明
0c	最初の書き込み
0d	バースト書き込み
09	最初の読み出し
0a	バースト読み出し
0e	書き込み待ち
03	プリチャージ
04	プリチャージ待ち
07	アクティブ
08	アクティブ待ち
16	書き込みから読み出しへ
0b	読み出し待ち

コントローラのインプリメント

コントローラ ステート マシンは、メモリのタイミング要件を考慮しながら適切な順序でコマンドを発行します。図 7 に、コントローラ ステート マシンの各段階を示します。



X702_07_021606

図 7: DDR2 コントローラ ステート マシン

コントローラからメモリにコマンドを発行する前の動作は次のとおりです。

1. コントローラにより、書き込みまたは読み出しアドレス FIFO に読み出しイネーブル信号が送信されます。
2. すべてのバンクがプリチャージされている場合は対応するバンクの行がアクティブになり、既にアクティブな行がある場合は、その行とバンクのアドレスが新しい行とバンクのアドレスと比較されます。異なる場合は、現在アクティブな行がプリチャージされ、読み出し/書き込みステートに遷移する前に **ACTIVE** コマンドが発行されます。
3. 書き込みステートのときにコントローラが **READ** コマンドを検出すると、**write_to_read** タイム後に **READ** コマンドが発行されます。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 **WRITE** コマンドが検出されると、**read_to_write** タイム後に **WRITE** コマンドが発行されます。

4. バックエンド ユーザー アプリケーションから PRECHARGE、AUTO-REFRESH、ACTIVE または LOAD MODE REGISTER コマンドが要求された場合、コントローラが PRECHARGE コマンドを発行します。
5. コマンドは、DDR2 メモリに対して発行される前に、パイプライン化されてアドレス信号に同期化されます。

デザイン階層

図 8に、最上位モジュール ddr2 以下のデザイン階層を示します。

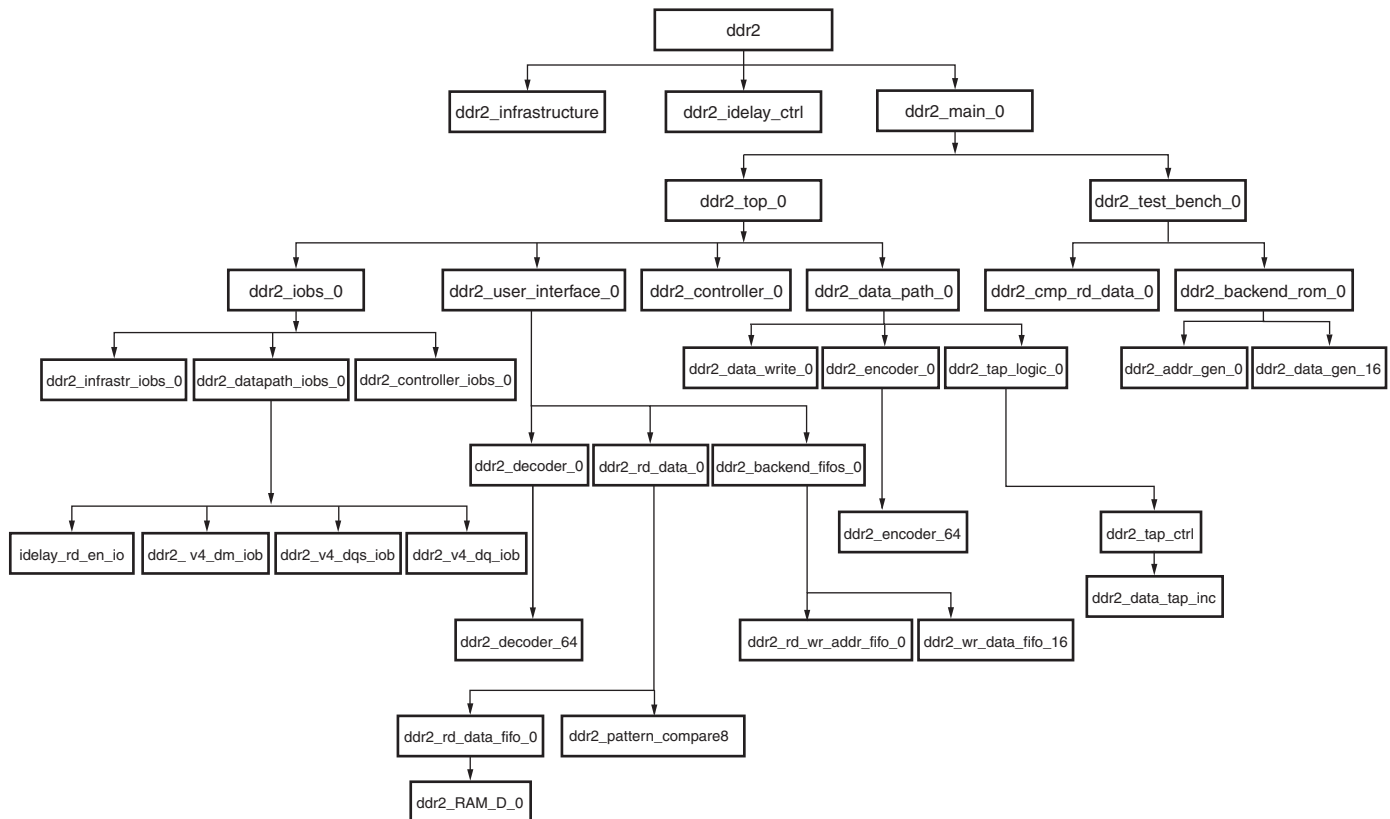


図 8：デザイン階層

リファレンス デザイン

ダイレクト クロッキング データ キャプチャ手法を使用した DDR2 SDRAM メモリ コントローラのリファレンス デザインは、Memory Interface Generator (MIG) ツールに統合されており、このツールは、ザイリンクス CORE Generator™ に統合されています。最新バージョンのデザインは、次のサイトから ISE IP アップデートをダウンロードすることにより入手できます。

http://japan.xilinx.com/xlnx/xil_sw_updates_home.jsp

終わりに

リファレンス デザインでは、DDR2 コントローラが適切な速度で動作するように、Virtex-4 DCM、IOB および差動クロック ツリーを使用しています。

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	改訂内容
2004/09/10	1.0	初版リリース
2005/06/13	1.1	「DDR2 インターフェイスのインプリメント」を改訂。
2005/07/15	1.2	表 4 のポート名を変更。「ユーザーからコントローラへのインターフェイス」を改訂。図 4 および 図 8 を変更。
2005/07/29	1.3	図 4 を変更およびリファレンス デザイン ファイルを更新。
2005/09/14	1.4	図 1 および「リファレンス デザイン」を改訂。
2005/11/18	1.5	「AUTO-REFRESH コマンド」を改訂。
2006/02/22	1.6	図 4、図 7 および 図 8 を改訂。 表 4、表 7、表 8 および 表 9 を改訂。
2007/03/01	1.7	<ul style="list-style-type: none">「はじめに」セクションを改訂。「拡張モード レジスタ」セクションの「初期化シーケンス」内の文を変更。図 8 を変更。 「リファレンス デザイン」セクションを改訂。
2007/04/23	1.8	<ul style="list-style-type: none">図 3 を復活。正しいタイミング関係に修正。図 5 の見出しを修正。